

EKONOMISK BELASTNINGSFÖRDELNING I  
KRAFTSYSTEM

J.RAGNARSSON



J. RAGNARSSON

EKONOMISK BELASTNINGSFÖRDELNING I KRAFTSYSTEM.

AV

JAHN RAGNARSSON

ABSTRACT.

Principiellt: Minimera  $f(x)$  under bivillkoren  $g(x) \leq 0$  o  $h(x) = 0$ .  
En översikt över metoder för lösning av detta problem ges.  
Hackdelen med de kända metoderna är att svårigheter uppstår då  
olikhetsvillkoren ej är uttryckta i de oberoende variablerna.  
En ny metod, ELD, som klarar av detta problem presenteras. Ett  
testexempel utarbetat av CIGRE används slutligen för att under-  
söka om o hur ELD fungerar.

In point of principle: Minimize  $f(x)$  if following conditions  
shall be satisfied  $g(x) \leq 0$  and  $h(x) = 0$ .

A summary of methods solving this problem is presented.  
The disadvantage with known methods is that difficulties comes  
if inequality constraints not are expressed in the independent  
variables.

A new method, ELD, which manage this problem is presented.  
A testexample worked out by CIGRE is finally used to test if  
and how ELD works.

<u>INNEHÅLLSFÖRTECKNING.</u>	<u>SIDA.</u>
INLEDNING	1
Kostnadsfunktionens utseende	1
Bivillkor	1
Knutpunktsvariabler	2
NÅGRA METODER FÖR MINIMERING UNDER BIVILLKOR	3
Problemmformulering	3
Minimering utan hänsyn till bivillkor	3
Minimering då hänsyn tas till likhetsvillkor	3
Minimering då hänsyn tas till både olikhets- o likhetsvillkor	3
Minimering m.h.a. numeriska metoder	4
ELD	7
Princip för metoden	7
Kommentarer till flödesschemat	8
TESTEXEMPEL	12
Resultat av minimeringen	15
SLUTSATSER	17
REFERENSER	18
APPENDIX	20
Huvudprogrammet MAIN	21
Subrutinen CINITG	26
Subrutinen FUNC	29
Subrutinen ADJINPL	36

INLEDNING.

Det ekonomiska belastningsfördelningsproblemet kan formuleras som problemet att minimera de momentana produktionskostnaderna samtidigt som kraftsystemets bivillkor skall vara uppfyllda.

Kostnadsfunktionens utseende.

Kostnaden att producera den aktiva effekten P bestäms av sambandet.

$$f(P) = \sum_{i=1}^{NG} c_i(PGEN_i)$$

där  $c_i$  är kostnadsfunktionen för generator nummer i av de NG som är i drift.

Bivillkor.

- a) Likhetsvillkor: Vid varje knutpunkt skall gälla att in i nätet injicerad effekt plus begärd effekt skall vara lika med den vid knutpunkten genererade effekten.
- b) Olikhetsvillkor: I leveranskontrakt anges det ofta att effekten skall levereras vid en spänning som tillåts variera inom bestämda gränser. "Toleransområdet" brukar vara spänningens nominella värde  $\pm 5$  eller  $\pm 10$  procent. Dessutom finns en övre gräns som beror på isoleringen. För spänningens absolutbelopp, VABS, skall alltså gälla vid var o en av de NB knutpunkterna att:

$$VMIN_i \leq VABS_i \leq VMAX_i$$

Under normal drift tillåts genererad aktiv effekt, PGEN, att variera mellan en övre o en undre gräns bestämd av restriktioner på den drivande turbinen. Övre gränsen kan också påverkas av en viss "spinning reserv". Den maximala reaktiva effekten vid maximal aktiv effekt bestäms av generatoren. Den minimala reaktiva effekten beror på förmågan hos

generatoren att konsumera reaktiv effekt utan att förlora synkroniseringen. För var o en av de NG generatorerna i drift i systemet skall gälla att:

$$P_{MIN,i} \leq P_{GEN,i} \leq P_{MAX,i}$$

$$Q_{MIN,i} \leq Q_{GEN,i} \leq Q_{MAX,i}$$

#### Knutpunktsvariabler.

För varje knutpunkt i ett kraftsystem finns fyra variabler: Spänningens real o imaginärdel (eller absolutbelopp o fasvinkel) samt in i nätet injicerad aktiv o reaktiv effekt. Eftersom det för varje knutpunkt finns två nätekvationer måste man välja två av dem som oberoende o de båda andra blir då beroende variabler.

## NÅGRA METODER FÖR MINIMERING UNDER BIVILLKOR.

### Problemformulering.

Det mest generella problemet kan formuleras:

$$\begin{aligned} & \text{minimera } f(x) \text{ under bivillkoren} \\ & g_i(x) \leq 0 \quad i=1 \dots m \\ & h_j(x) = 0 \quad j=1 \dots p \end{aligned} \quad (1)$$

där  $x$  är en  $n$ -dimensionell vektor.

För olika specialfall har man utarbetat olika metoder.

### Minimering utan hänsyn till bivillkor ( $m=p=0$ ).

Problemet är reducerat till att minimera  $f(x)$ . Ett nödvändigt men inte tillräckligt villkor för minimum är de ekvationer som definieras av:

$$\frac{\partial f(x)}{\partial x} = 0 \quad (2)$$

Ett stort antal minimeringsmetoder som effektivt löser detta problem har utarbetats [1]-[5].

### Minimering då hänsyn tas till likhetsvillkor ( $p>0, m=0$ ).

Det klassiska sättet att lösa sådana problem är m.h.a.

Lagranges multiplikatorer. Följande ekvationer skall i detta fall vara uppfyllda:

$$\begin{aligned} & h_j(x) = 0 \quad j=1 \dots p \\ & \frac{\partial f(x)}{\partial x} + \sum_{j=1}^p \lambda_j \frac{\partial h_j(x)}{\partial x} = 0 \end{aligned} \quad (3)$$

### Minimering då hänsyn tas till både olikhets- o likhetsvillkor ( $p>0, m>0$ ).

En utvidgning av metoden med Lagranges multiplikatorer introducerades av Kuhn-Tucker [7]. Metoden kräver lösning av:

$$\begin{aligned} & h_j(x) = 0 \quad j=1 \dots p \\ & \frac{\partial f(x)}{\partial x} + \sum_{j=1}^p \lambda_j \frac{\partial h_j(x)}{\partial x} + \sum_{i=1}^m U_i \frac{\partial g_i(x)}{\partial x} = 0 \\ & U_i g_i(x) = 0 \quad i=1 \dots m \\ & U_i \geq 0 \quad i=1 \dots m \end{aligned} \quad (4)$$

Dessa ekvationer är ofta mycket svåra att lösa.

Minimering m.h.s. numeriska metoder.

De metoder som här kommer att diskuteras baserar sig på att man kan transformera det ursprungliga problemet (1) till ett problem utan bivillkor, vars minimum sammanfaller med det ursprungliga problemets. Skälet till att man transformerar problemet är att effektiva metoder finns för minimering av problem utan bivillkor [1]-[5]. Carrol [8] föreslog en metod som Fiacco-McCormick [9]-[11] senare utvecklade. Den löser (1) då  $m > 0$  o  $p = 0$ . Metoden går ut på att minimera funktionen:

$$P(x, r_k) = f(x) + r_k \sum_{i=1}^m \frac{1}{g_i(x)} \quad (5)$$

Man startar med ett begynnelsevärde på  $x$  som uppfyller alla olikhetsvillkoren o med ett specialvärde på konstanten  $r_1$ . Här man funnit minimum av  $P(x, r_1)$  väljes  $r_2 < r_1$  o  $P(x, r_2)$  minimeras. Då  $k$  blivit tillräckligt stort sammanfaller minimum för  $P(x, r_k)$  med minimum för  $f(x)$ . Metoden kallas SUMT (Sequential Unconstrained Minimization Technique).  $1/g$  termens värde ökar snabbt då  $g_i(x)$  närmar sig sin gräns noll o har därmed en "repellerande" inverkan. Fiacco-McCormick [10] föreslog att  $r_1$  skulle bestämmas av sambandet:

$$r_1 = - \frac{\sum_{j=1}^n \frac{\partial f}{\partial x_j} \cdot \frac{\partial e}{\partial x_j}}{|\partial e / \partial x|^2} \quad (6)$$

där 
$$e = \sum_{i=1}^m \frac{1}{g_i(x)}$$

En förutsättning är att  $r_1$  är större än noll. Om inte reducerar man  $f(x)$  genom att ta steg av given längd i riktning som ges av negativa gradienten till  $f(x)$  o beräknar  $r_1$  efter varje steg tills ett positivt värde erhållits.

En olägenhet med Fiacco-McCormicks metod är att värdet på  $f(x)$  har ett maximalt fel som ges av:

$$E(x, r_k) = r_k \sum_{i=1}^m \frac{1}{g_i(x)} \quad (7)$$

då  $P(x, r_k)$  minimerats. Eftersom slutvärdena på  $g_i(x)$  inte är kända då minimeringsprocessen börjar kan (7) bara användas för att kontrollera metodens konvergens.



Lootsma [12] sökte efter en funktion av  $g_i(x)$  som skulle ge ett fel hos  $f(x)$  oberoende av  $g_i(x)$ . En funktion vars avvikel-  
se från  $f(x)$  kan uttryckas som:

$$E(x, r_k) = \delta \quad (8)$$

$$\text{är } L(x, r_k) = f(x) - r_k \sum_{j=1}^m \ln(-g_j(x)) \quad (9)$$

under förutsättning att  $\delta$  väljes som:

$$\delta = m \cdot r_k \quad (10)$$

Att man kan specificera  $\delta$  medför att  $f(x)$  i minimumpunkten kan approximeras med en på förhand bestämd noggrannhet.

Fiacco-McCormick o Lootsmas metoder kräver ett startvärde på  $x$  som medför att alla olikhetsvillkoren är satisfierade, vilket är en mycket stark begränsning för många problem. Ett sätt att snabbt erhålla en godtycklig eller specificerad startpunkt är att använda följande funktion:

$$F(x) = \sum_{i=1}^m (g_i(x) - b_i)^2 \quad (11)$$

där  $b_i$  är negativa konstanter.

$F(x)$  är en funktion utan bivillkor o kan alltså minimeras direkt.  $F(x)$ 's minimivärde är noll. Detta erhålles då varje  $g_i(x)$  är lika med motsvarande  $b_i$ . Om bara en punkt som ligger i den rymd som begränsas av olikhetsvillkoren önskas kan minimeringen avbrytas då alla  $g_i(x) \leq 0$ . Olikhetsvillkoret  $g_i(x) \leq 0$  behandlas i (11) som ett likhetsvillkor. Ekv. (1) kan alltså för  $m=0$  o  $p>0$  transformeras till minimering av:

$$P(x) = f(x) + \sum_{j=1}^p (h_j(x))^2 \quad (12)$$

Minimering av (12) är tillräcklig för att erhålla minimum hos  $f(x)$  under förutsättning att man inte når ett tillstånd i minimeringsprocessen då en variation i  $x$  för att reducera  $\sum$ -termen medför en lika stor ökning av  $f(x)$ . Under sådana omständigheter skulle minimeringsprocessen finna att  $P(x)$  hade nått minimum i en punkt där likhetsvillkoren ej är uppfyllda.

För att hindra att ovannämnda minimipunkt erhålles modifieras (12) till:

$$P(x, r_k) = f(x) + \frac{1}{r_k} \sum_{j=1}^p (h_j(x))^2 \quad (13)$$

Lägg märke till att med  $r_k=1$  är (13) o (12) identiska. Efter minimering av  $P(x, r_1)$  med  $r_1=1$  kan  $r$  ändras till  $r_2 < r_1$  o minimeringsprocessen repeteras. Minimeringen avbrytes då  $\Sigma$ -termen har reducerats till noll.

K.Mårtensson [6] har presenterat en metod som kan sägas vara en kombination av (3) o (13). Han inför begreppet multiplikatorfunktion för vektorfunktionen  $\mu(x)$  och hans "generaliserade Lagrangefunktion" är:

$$H(x, c) = f(x) + \mu^T(x) h(x) + c h^T(x) h(x) \quad (14)$$

Fiacco-McCormick [13] presenterade senare en metod som utvidgade deras tidigare metod till att även gälla problem med likhets- o olikhetsvillkor. De föreslog att man adderade en term till (13) som tar hänsyn till olikhetsvillkoren.

$$P(x, r_k) = f(x) + r_k \sum_{i=1}^m \frac{1}{g_i(x)} + \frac{1}{r_k} \sum_{j=1}^p (h_j(x))^2 \quad (15)$$

Den stora nackdelen med (15) är att det är svårt att uppfylla kravet på en punkt som medför att olikhetsvillkoren är satisfierade.

Fiacco-McCormick [14] och Zangwill [15] modifierade därför (13) på ett annat sätt. De föreslog väsentligen samma alternativ.

$$P(x, r_k) = f(x) + \frac{1}{r_k} \sum_{j=1}^p (h_j(x))^2 + \frac{1}{r_k} \sum_{i=1}^m (g_i(x) - b_i)^2$$

$$b_i = \begin{cases} g_i(x) & \text{om } g_i(x) \leq 0 \\ 0 & \text{om } g_i(x) > 0 \end{cases} \quad (16)$$

Ovannämnda metod kräver ingen startpunkt som ligger i den rymd som begränsas av bivillkoren vilket är en klar fördel. Zangwill visade att man under vissa omständigheter kan nå minimum efter en minimeringsprocess o då behövs bara ett värde på  $r$ .

Praktiskt stoppas processen då  $P(x, r_k)$  har minimerats o minimumvärdet är tillräckligt nära värdet på  $f(x)$ .

ELD.

ELD (Economic Load Dispatch program) är ett förslag till lösningsmetod som använder sig av knutpunktsspänningarnas real o imaginärdel som oberoende variabler.

Princip för metoden:

Metoden innebär en utvidgning av K.Mårtenssons [6] till att även gälla olikhetsvillkor. Vid den första minimeringen används den utökade funktionen:

$$F(x, \lambda) = f(x) + \lambda^T h(x) + C h^T(x) h(x) + D g_I(x)$$

$$\text{där } g_{IIi}(x) = \begin{cases} g_i(x)^3 & \text{om } g_i(x) > 0 \\ 0 & \text{om } g_i(x) \leq 0 \end{cases}$$

Denna funktion minimeras med fixa värden på  $\lambda$ , C o D. Om resultatet av denna minimering blir att det för något i gäller att  $g_i(x) > 0$  behandlas  $g_i(x)$  vid nästa minimering som ett likhetsvillkor o benämnes aktivt olikhetsvillkor. I fortsättningen minimeras alltså:

$$F(x, \lambda, \mu) = f(x) + \lambda^T h(x) + C h^T(x) h(x) + D g_I(x) + \mu^T g_{II}(x) + D g_{II}^T(x) g_{II}(x)$$

$$\text{där } g_{IIIi}(x) = \begin{cases} g_i(x) & \text{om } g_i(x) > 0 \text{ och olikhetsvillkoret} \\ & \text{aktivt} \\ 0 & \text{för övrigt} \end{cases}$$

Med hjälp av de tre heltalsvektorer IPV(\*), IQV(\*) o IVV(\*), som uppdateras efter varje minimering, håller man reda på om olikhetsvillkoren är aktiva eller inaktiva. Om resultatet av föregående minimering blev att för något i t.ex. en maximumgräns överskridits sätts motsvarande element i vektorn =+1. Om i stället en mingräns underskridits sätts motsvarande element =-1. Exempel: Om  $PGM(8) < PML(8)$  sättes  $IPV(8) = -1$ . Från början är samtliga element i vektorena lika med noll. På de närmast följande sidorna presenteras flödesschemat samt kommentarer till detta.

Kommentarer till flödesschemat.1. LÄS INDATA.

Följande indata om kraftsystemet behövs till ELD:

Låt  $z(*)$  beteckna att  $z$  är en vektor.

NB antalet knutpunkter.  
 NG antalet generatorer.  
 NL antalet ledningar.  
 $a1(*)$ ,  $a2(*)$  som kostnadsfunktion för generator  $i$ ,

$c_i$ , har i denna algoritmen valts ett andragsgradspolynom i genererad aktiv effekt:

$$c_i(PGEN_i) = a0_i + a1_i PGEN_i + a2_i PGEN_i^2$$

Detta har visat sig vara en god approximation i området:  $PMIN_i \leq PGEN_i \leq PMAX_i$ .

Eftersom  $a0_i$  inte har någon betydelse för minimeringen utelämnas denna.

$PMIN(*)$ ,  $PMAX(*)$ ,

$QMIN(*)$ ,  $QMAX(*)$

begränsningar på genererad aktiv resp. reaktiv effekt vid de olika generatorerna.

$NGEN(*)$

heltalsvektor som talar om antalet generatorer vid varje knutpunkt.

$PD(*)$ ,  $OD(*)$

begärd aktiv resp. reaktiv effekt för varje knutpunkt.

$VMIN(*)$ ,  $VMAX(*)$

begränsningar på knutpunktsspänningen

$GAA(*)$ ,  $BAA(*)$

real resp. imaginärdel av shuntadmittansen vid knutpunkt A.

$RAB(*)$ ,  $XAB(*)$

real resp. imaginärdel av impedansen hos ledningen mellan A o B.

$GBB(*)$ ,  $BBB(*)$

real resp. imaginärdel av shuntadmittansen vid knutpunkt B.

$LTA(*)$

ändpunkt A på ledningen.

$LTB(*)$

ändpunkt B på ledningen.

$EPSP$ ,  $EPSQ$

testkvantitet för likhetsvillkor.

$EPSFG$ ,  $EPSQG$ ,  $EPSV$

testkvantitet för olikhetsvillkor.

$C$ ,  $D$

koefficienter som används i  $F(x, \lambda, \mu)$ .

## 2. SKRIV INDATA.

För att kontrollera att rätt data stansats o lästs in skrivs alla indata ut.

## 3. BERÄKNA INITIALVÄRDEN.

För att minimeringen skall gå så snabbt som möjligt vill man vid inträdet till minimeringsprocessen ha en så god uppskattning av minimumpunkten som möjligt. Initialvärdena beräknas i subrutinen CINITG (Compute INITIAL Guess) o sker där enligt följande:

Antag att totala behovet av aktiv effekt är  $PDEM = \sum_{i=1}^{NG} PD_i$ . Om inga begränsningar finns på PGEN o ledningsförlusterna var lika med noll skulle fördelningen av genererad aktiv effekt på de olika generatorerna, vid minimal produktionskostnad under bivillkoret  $\sum_{i=1}^{NG} PGEN_i = PDEM$ , vara sådan att  $dc_i/dPGEN_i$  var lika för samtliga generatorer. Om vi nu tar hänsyn till begränsningarna genom att sätta den eller de  $PGEN_i$  som når en gräns lika med gränsen, skall för de övriga gälla liksom ovan att  $dc_i/dPGEN_i$  är lika. Ovanstående resonemang finns utförligare hos Elgerd [16]. En subrutin som utför dessa beräkningar ELDNL (Economic Load Dispatch Neglecting transmissions Losses) fördelar i första steget produktionen av aktiv effekt på de olika generatorerna när man sätter  $PDEM = \sum_{i=1}^{NB} PD_i$ . Därefter användes LFLOW (Computes a solution to the load-flow problem using the Newton-Raphson method with cartesian and polar coordinates). Indata till subrutinen är två av de fyra knutpunktvariablerna. LFLOW beräknar då de andra två. Knutpunkt 1 användes som utjämningsknutpunkt. Med hjälp av resultatet från LFLOW beräknas ledningsförlusterna i systemet PLOSS. ELDNL anropas därefter på nytt, nu med  $PDEM = PLOSS + \sum_{i=1}^{NB} PD_i$ . Slutligen anropas LFLOW på nytt.

## 4. UPPSKATTNING AV MULTIPLIKATORER.

För likhetsvillkoren används multiplikatorerna  $PLA(*)$  o  $QLA(*)$ , som tillsammans utgör vektorn  $\lambda$ .

För olikhetsvillkoren används multiplikatorerna  $BEY(*)$ ,  $OMEY(*)$  o  $VMY(*)$ , som tillsammans utgör vektorn  $\mu$ .

Alla  $PLA_i$  sättes lika med det värde som ELDNL ger på  $dc_i/dPGEN_i$ . De övriga multiplikatorerna sättes lika med noll.

### 5. MINIMERING M.H.A. FIXA MULTIPLIKATORER.

$F(x, \lambda, \mu)$  minimeras här m.h.a. subrutinen FLEET, som arbetar enligt Fletcher-Powells metod [1] modifierad av Broyden, Shanno o Fletcher [2] - [5]. Metoden utnyttjar funktionsvärde o gradient vilka beräknas i subrutinen FUNC.

### 6. ALLA VILLKOR UPPEFYLLDA?

Följande krav ställs på bivillkoren:

$$|h_j(x)| \leq \text{EPSP eller EPSQ}$$

$$g_1(x) \leq \text{EPS}$$

där EPS står för EPSPG, EPSQG eller EPSV.

I ELD har följande testkvantiteter använts:

EPSP=0,01 MW

EPSQ=0,01 MVA

EPSPG=0,0001 MW

EPSQG=0,0001 MVA

EPSV=0,0001 kV

### 7. ÄNDRA MULTIPLIKATORERNA.

- a) Likhetsvillkoren : Multiplikatorerna PLA(\*) o QLA(\*) ändras här liksom i K.Mårtenssons metod [6], med en faktor proportionell mot avvikelserna.
- b) Olikhetsvillkoren: Ändringarna i PBY(\*), QBY(\*) o VBY(\*) beror på om olikhetsvillkoret var aktivt eller inaktivt under föregående minimering (avgöres m.h.a. IPV(\*), IQV(\*) o IVV(\*)). Om det var aktivt o fortfarande ej är uppfyllt ändras multiplikatorerna med en faktor proportionell mot  $D$  o  $|g_1(x)|$ . Var det däremot inaktivt under minimeringen men ej uppfyllt efter minimeringen sätts multiplikatorerna lika med en kvantitet proportionell mot  $D$  o  $|g_1(x)|^2$ . Efter ändringen av multiplikatorerna uppdateras de tre vektorerna IPV(\*), IQV(\*) o IVV(\*).

Samtliga ändringar sker i subrutinen ADJMULT (ADJUST MULTIPLIERS).

### 3. SKRIV RESULTAT.

En lång rad data skrives ut bl.a. knutpunktsspänningar, genererad aktiv effekt vid de olika generatorerna, injicerad aktiv o reaktiv effekt vid de olika knutpunkterna, värde på multiplikatorer samt kostnadsfunktionens värde vid minimum. Exempel på resultatutskrift finns presenterad i anslutning till testexemplet.

I appendix finns listor på huvudprogrammet MAIN samt subrutinerna CINITG, FUNC o ADJPL.

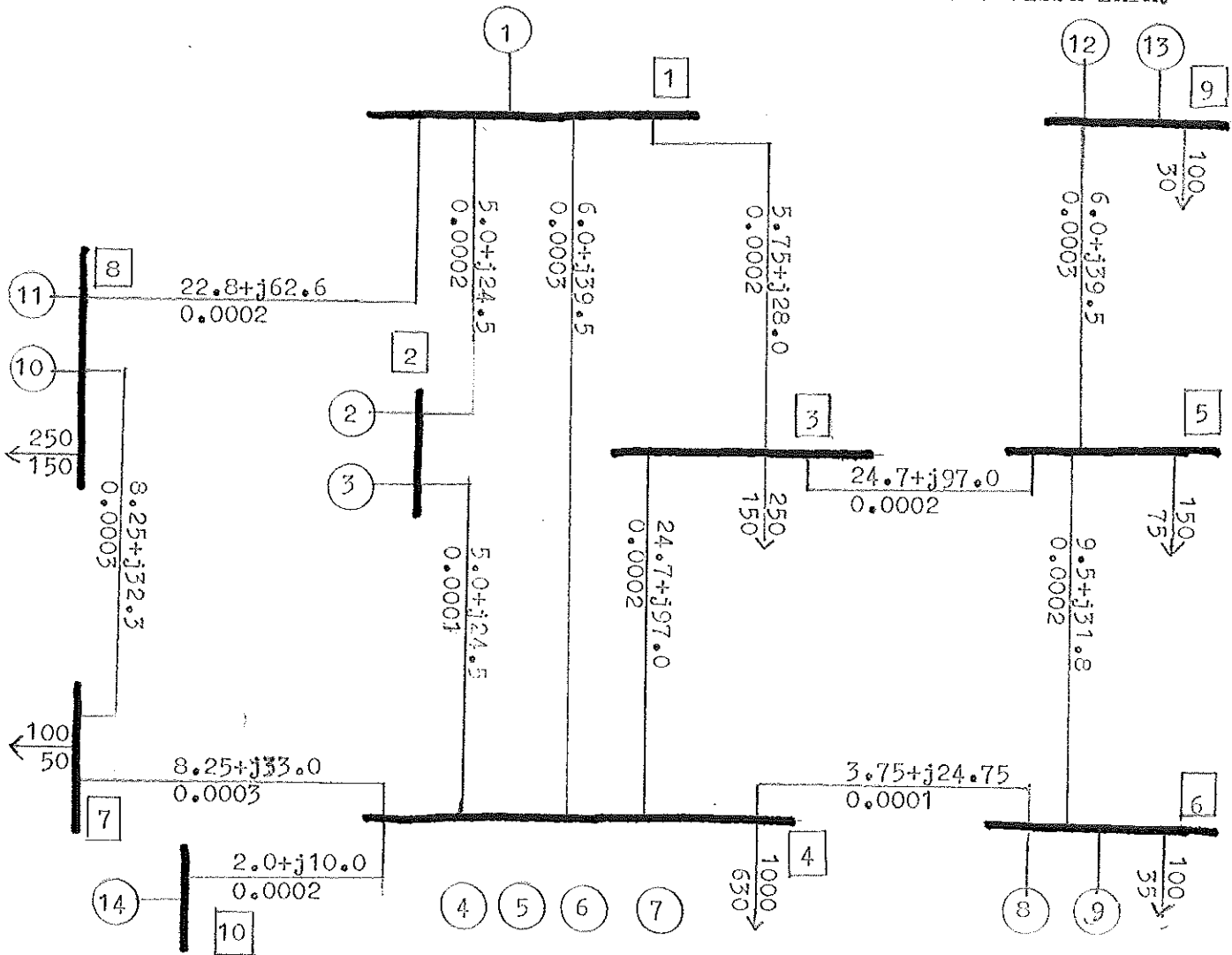
TESTEXEMPEL.

För att testa om o hur ELD fungerade användes den på ett test-exempel utarbetat av CIGRE (Conference Internationale des Grands Reseaux Electriques) [17]. Kraftsystemets uppbyggnad o data framgår av de närmast följande sidorna. Kostnaden är given i vissa punkter. För varje generator har m.h.a. dessa punkter gjorts en anpassning av ett andragradspolynom.



# KRAFTSYSTEMET

UPPBYGGNAD O VISSA DATA.



## BETECKNINGAR:

$\boxed{K}$

K=Knutpunktsnummer

$\odot G$

G=Generatormummer

$\frac{PD}{QD} \rightarrow$

Knutpunktsbelastningar. PD i MW o QD i MVar.

$\frac{RAB+jXAB}{BAA=BBB}$

$RAB+jXAB$  ledningsimpedens i ohm.  $BAA=BBB$  imaginär-del av shuntadmittans i mho.

# 14 UTSKRIFT AV INDATA

INPUT DATA TO ECONOMIC LOAD DISPATCH PROGRAM  
 \*\*\*\*\*

NUMBER OF BUSES = 10  
 NUMBER OF GENERATORS = 14  
 NUMBER OF LINES = 13  
 COEFFICIENT IN COST FUNCTION C = 5.0  
 COEFFICIENT IN COST FUNCTION D = 10.0

GEN.NR	A1	A2	PMIN	PMAX	QMIN	QMAX
1	.773	.00340	80.0	217.0	-24.0	120.0
2	.395	.00443	80.0	217.0	-24.0	120.0
3	.538	.00407	80.0	217.0	-24.0	120.0
4	.768	.00350	80.0	217.0	-24.0	120.0
5	.623	.00384	80.0	217.0	-24.0	120.0
6	.636	.01008	40.0	108.0	-15.0	75.0
7	.696	.00978	40.0	108.0	-15.0	75.0
8	.675	.01033	40.0	108.0	-15.0	75.0
9	.803	.00966	40.0	108.0	-15.0	75.0
10	.394	.00392	80.0	217.0	-24.0	120.0
11	1.367	.00623	40.0	108.0	-15.0	75.0
12	1.513	.00602	30.0	72.0	-8.0	40.0
13	.678	.00773	40.0	108.0	-15.0	75.0
14	.489	.00393	80.0	217.0	-24.0	120.0

BUS.NR	NGEN	PD	QD	VMIN	VMAX
1	1	.0	.0	205.0	240.0
2	2	.0	.0	205.0	240.0
3	0	250.0	150.0	205.0	240.0
4	4	1000.0	630.0	205.0	240.0
5	0	150.0	75.0	205.0	240.0
6	2	100.0	35.0	205.0	240.0
7	0	100.0	50.0	205.0	240.0
8	2	250.0	150.0	205.0	240.0
9	2	100.0	30.0	205.0	240.0
10	1	.0	.0	205.0	240.0

LINE.NR	GAA	BAA	RAB	XAB	GBB	BBB	LTA	LTB
1	.0	.0002	5.00	24.50	.0	.0002	1	2
2	.0	.0002	5.75	28.00	.0	.0002	1	3
3	.0	.0003	6.00	39.50	.0	.0003	1	4
4	.0	.0001	5.00	24.50	.0	.0001	2	4
5	.0	.0002	24.70	97.00	.0	.0002	3	4
6	.0	.0002	24.70	97.00	.0	.0002	3	5
7	.0	.0001	3.75	24.75	.0	.0001	4	6
8	.0	.0003	8.25	33.00	.0	.0003	4	7
9	.0	.0002	2.00	10.00	.0	.0002	4	10
10	.0	.0002	9.50	31.80	.0	.0002	5	6
11	.0	.0003	6.00	39.50	.0	.0003	5	9
12	.0	.0003	8.25	32.30	.0	.0003	7	8
13	.0	.0002	22.80	62.60	.0	.0002	8	1

Resultat av minimeringen.

I nedanstående tabell anges det antal funktionsevalveringar som erfordras för att nå minimum för olika värden på konstanterna C o D.

D \ C	1	10	20
1	211	> 300	> 300
5	> 300	83	> 300
10	> 300	> 300	> 300

Tabell 1.

C=5 o D=10 ger det snabbaste resultatet (tid för körning av hela programmet 8,062 s) och även det lägsta värdet på kostnadsfunktionen (2785,636).

Totalt produceras i detta fall 1971,3834 MW.

Eftersom det sammanlagda behovet är 1950,0000 MW blir ledningsförlusterna 21,3834 MW.

Resultatutskrift från minimeringen med C=5 o D=10 finns på nästa sida.

# RESULTATUTSKRIFT

## RESULT FROM ECONOMIC LOAD DISPATCH PROGRAM

\*\*\*\*\*

BUS	PINJ	QINJ	VMIN	VABS	VMAX	VANG
1	204.0816	119.7097	205.0000	237.6229	240.0000	-.0137
2	386.6647	44.4841	205.0000	240.0026	240.0000	.0429
3	-249.9983	-149.9985	205.0000	219.3668	240.0000	-.1214
4	-432.3467	-240.3737	205.0000	229.7383	240.0000	-.0650
5	-149.9977	-75.0003	205.0000	227.8595	240.0000	-.1331
6	48.4780	56.5331	205.0000	234.3043	240.0000	-.0858
7	-99.9997	-49.9999	205.0000	228.1123	240.0000	-.0861
8	33.9833	-12.6996	205.0000	232.9842	240.0000	-.0486
9	64.8469	27.4735	205.0000	236.6865	240.0000	-.0905
10	215.6755	83.3046	205.0000	235.4277	240.0000	-.0286

GEN.NR	PMIN	PGEN	PMAX
1	80.0000	204.0816	217.0000
2	80.0000	193.5559	217.0000
3	80.0000	193.1088	217.0000
4	80.0000	205.8221	217.0000
5	80.0000	206.4784	217.0000
6	40.0000	78.0136	108.0000
7	40.0000	77.3392	108.0000
8	40.0000	74.9524	108.0000
9	40.0000	73.5257	108.0000
10	80.0000	217.0000	217.0000
11	40.0000	66.9833	108.0000
12	30.0000	62.3103	72.0000
13	40.0000	102.5366	108.0000
14	80.0000	215.6755	217.0000

### MULTIPLIERS

BUS	PLA	QLA	IQV	QMY	IVV	VMY
1	2.1608	.0000	0	.0000	0	.0000
2	2.1099	.0000	0	.0000	1	.2121
3	2.2748	.0770	0	.0000	0	.0000
4	2.2088	.0000	0	.0000	0	.0000
5	2.2872	-.0206	0	.0000	0	.0000
6	2.2235	.0000	0	.0000	0	.0000
7	2.1839	.0006	0	.0000	0	.0000
8	2.2016	.0000	0	.0000	0	.0000
9	2.2632	.0000	0	.0000	0	.0000
10	2.1842	.0000	0	.0000	0	.0000

GENERATOR COST FUNCTION INKL. PENALTIES = 2785.646  
 GENERATOR COST FUNCTION EXKL. PENALTIES = 2785.636

SLUTSÄTTER.

Som framgick av tabell 1 är metodens konvergenshastighet starkt beroende av valet av  $C$  o  $D$ . För  $C=5$  o  $D=10$  är minimeringstiden mindre än 8 sekunder varför metoden för detta val av  $C$  o  $D$  får anses som snabb. Som vid all minimering av funktioner av många variabler kan man bara vara säker på att ett lokalt minimum erhållits. Om detta även är ett globalt minimum kan inte bevisas. Det är alltså möjligt att man med ett annat val av  $C$  o  $D$  eller en annan initialgissning kan få ett lägre värde på kostnadsfunktionen.

REFERENCES.

- [1] Fletcher, R., and Powell, M.J.D. : "A Rapidly Convergent Descent Method for Minimization", Computer Journal, Vol.6, pp 163-168, 1963.
- [2] Broyden, C.G. : "Quasi-Newton Methods and their Application to Function Minimization", Math.Comp.Vol.21, pp 368-381, 1967.
- [3] Shanno, D.F. : "Inverse Quasi-Newton Methods", Center for Mathematical Studies, Report 6938, University of Chicago, 1969.
- [4] Broyden, C.G. : "The Convergence of Single-Rank Quasi-Newton Methods, Math.Comp.Vol.24, 1970.
- [5] Fletcher, R. : "A New Approach to Variable Metric Algorithms", Comp.J. , Vol.13, pp 317-322, 1970.
- [6] Mårtensson, K. : "A New Approach to Constrained Function Optimization" Report 7112 september 1971 Lund Institute of Technology Division of Automatic Control.
- [7] H.W. Kuhn and A.W. Tucker : "Nonlinear programming", Proc. 2nd Berkeley Symp. on Mathematical Statistics and Probability, pp 481-492, 1951.
- [8] C.W. Carrol : "The created response surface technique for optimizing nonlinear restrained systems", Operations Res. vol.9, pp 169-184, 1961.
- [9] A.V. Fiacco and G.P. McCormick : "The Sequential unconstrained minimization technique for nonlinear programming, a primal-dual method" Management Sci. vol.10, pp 360-366, 1964.
- [10] A.V. Fiacco and G.P. McCormick : "Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming" Management Sci. vol.10, pp 601-617, 1964.
- [11] A.V. Fiacco and G.P. McCormick : "Programming under nonlinear constraints by unconstrained minimization, a primal-dual method" Research Analysis Corp., TP-96, 1963.

- [12] E.A. Lootsma : "Logarithmic programming: a method of solving nonlinear-programming problems" Phillips Res. Repts., vol.22, pp 329-344.
- [13] A.V. Fiacco and G.P. McCormick : "The sequential unconstrained minimization technique for convex programming with equality constraints" Research Analysis Corp. TP-155, 1965.
- [14] A.V. Fiacco and G.P. McCormick : "The Slacked unconstrained minimization technique for convex programming" SIAM J. Appl.Math., vol.15, pp 505-515, 1967.
- [15] W.I. Zangwill : "Non-linear programming via penalty functions" Management Sci., vol.13, pp 344-358, 1967.
- [16] Oile I Elgerd : "Electric energy systems theory: an introduction" Chapter 8, 1971.
- [17] CIGRE 32-19, "Optimization studies for the operation of power systems", 1972.

# APPENDIX



# HUVUDPROGRAMMET, MAIN

```

1*      C      *****
2*      C      * ECONOMIC LOAD DISPATCH PROGRAM *
3*      C      *****
4*      C      FOLLOWING PROBLEM IS SOLVED
5*      C      MINIMIZE THE COST FUNCTION IF VOLTAGE MAGNITUDE, GENERATED ACTIVE
6*      C      POWER AND GENERATED REACTIVE POWER ARE LIMITED AND THE LOAD-FLOW
7*      C      EQUATIONS SHALL BE SATISFIED
8*      C
9*      C
10*     C      AUTHOR, JAHN RAGNARSSON 9-4-1973
11*     C
12*     C      SUBROUTINE REQUIRED
13*     C
14*     C      CINITG
15*     C      ELDNL
16*     C      LFLOW
17*     C      CMAPRI
18*     C      DECOM
19*     C      MAPRI
20*     C      NORM
21*     C      SOLVB
22*     C      FUNC
23*     C      FRMIN
24*     C      ADJMP
25*     C
26*     C      START MAIN PROGRAM
27*
28*     C      INTEGER CR
29*     C      PARAMETER MB=10, MG=14, ML=13, MBB=20, MHH=MBB*(MBB+1)/2, MWW=4*MBB
30*     C      DIMENSION
31*     C      *QMIN(MG), QMAX(MG),
32*     C      *PMIN(MG), PMAX(MG), QQMIN(MB), QQMAX(MB), VMIN(MB), VMAX(MB),
33*     C      *PLA(MB), QLA(MB), PMY(MG), QMY(MB), VMY(MB), IPV(MG), IQV(MB), IVV(MB),
34*     C      *GAA(ML), BAA(ML), RAB(ML), XAB(ML), GBB(ML), BBB(ML), LTA(ML), LTB(ML),
35*     C      *DPDE(MB, MB), DPDF(MB, MB), DQDE(MB, MB), DQDF(MB, MB),
36*     C      *A1(MG), A2(MG), PGEN(MG), NGEN(MB), AINV(MB), NTYP(MB),
37*     C      *P(MB), PINJ(MB), PD(MB), Q(MB), QINJ(MB), QD(MB), VABS(MB), VANG(MB),
38*     C      *VB(MB), GO(MBB),
39*     C      *G(MBB),
40*     C      *X(MBB), H(MHH), W(MWW), EPSX(MBB),
41*     C      *Y(MB, MB), IB(MB)
42*     C      COMPLEX VB, IB, Y
43*     C      COMMON/BL 1/ NB, NG, NL, IPRINT
44*     C      COMMON/BL 2/ PMIN, PMAX, QQMIN, QQMAX, VMIN, VMAX
45*     C      COMMON/BL 3/ PLA, QLA, PMY, QMY, VMY, IPV, IQV, IVV, C, D
46*     C      COMMON/BL 4/ GAA, BAA, RAB, XAB, GBB, BBB, LTA, LTB
47*     C      COMMON/BL 5/ DPDE, DPDF, DQDE, DQDF
48*     C      COMMON/BL 6/ A1, A2, PGEN, NGEN, AINV, NTYP
49*     C      COMMON/BL 7/ P, PINJ, PD, Q, QINJ, QD, VABS, VANG, VB
50*     C      COMMON/BL 8/ G
51*     C      COMMON/BL 9/ X
52*     C      COMMON/BL10/ Y, IB
53*     C      COMMON/BL11/ FO, GO
54*
55*     C      DATA LP/6/, EPSPG/0.001/, EPSQG/0.001/,
      *EPSP/0.01/, EPSQ/0.01/, EPSV/0.01/, CR/5/

```

```

56*      DATA JJAC/1/,MAXIT/20/,EPS/1.0E-4/
57*      EXTERNAL FUNC
58*      C
59*      C *****
60*      C * READ INPUT DATA *
61*      C *****
62*      C
63*      READ(CR,10) NB,NG,NL,IPRINT
64*      READ(CR,20) C,D
65*      READ(CR,30) (A1(I),A2(I),PMIN(I),PMAX(I),QMIN(I),QMAX(I),I=1,NG)
66*      READ(CR,40) (NGEN(I),PD(I),QD(I),VMIN(I),VMAX(I),I=1,NB)
67*      READ(CR,50) (GAA(I),BAA(I),RAB(I),XAB(I),GBB(I),BBB(I),
68*      *LTA(I),LTB(I),I=1,NL)
69*      C
70*      10 FORMAT(4I5)
71*      20 FORMAT(2F5.1)
72*      30 FORMAT(F5.3,F10.5,2F5.2,2F5.1)
73*      40 FORMAT(I5,4F5.1)
74*      50 FORMAT(F5.1,F5.4,2F5.2,F5.1,F5.4,2I2)
75*      C
76*      C *****
77*      C * PRINT INPUT DATA *
78*      C *****
79*      WRITE(LP,200)
80*      WRITE(LP,210) NB,NG,NL,C,D
81*      WRITE(LP,220)
82*      DO 110 I=1,NG
83*      110 WRITE(LP,230) I,A1(I),A2(I),PMIN(I),PMAX(I),QMIN(I),QMAX(I)
84*      WRITE(LP,240)
85*      DO 120 I=1,NB
86*      120 WRITE(LP,250) I,NGEN(I),PD(I),QD(I),VMIN(I),VMAX(I)
87*      WRITE(LP,260)
88*      DO 130 I=1,NL
89*      130 WRITE(LP,270) I,GAA(I),BAA(I),RAB(I),XAB(I),GBB(I),BBB(I),
90*      *LTA(I),LTB(I)
91*      C
92*      200 FORMAT(1H1,9X,44HINPUT DATA TO ECONOMIC LOAD DISPATCH PROGRAM/
93*      *10X,44(1H*))
94*      210 FORMAT(///10X,17HNUMBER OF BUSES =,I5/
95*      *10X,22HNUMBER OF GENERATORS =,I5/
96*      *10X,17HNUMBER OF LINES =,I5/
97*      *10X,32HCoefficient IN COST FUNCTION C =,F5.1/
98*      *10X,32HCoefficient IN COST FUNCTION D =,F5.1)
99*      220 FORMAT(///10X,10H      GEN.NR,10H      A1,10H      A2,
100*      *10H      PMIN,10H      PMAX,10H      QMIN,10H      QMAX)
101*      230 FORMAT(10X,I10,F10.3,F10.5,4F10.1)
102*      240 FORMAT(///10X,10H      BUS.NR,10H      NGEN,10H      PD,
103*      *10H      QD,10H      VMIN,10H      VMAX)
104*      250 FORMAT(10X,2I10,4F10.1)
105*      260 FORMAT(///10X,10H      LINE.NR,10H      GAA,10H      BAA,
106*      *10H      RAB,10H      XAB,10H      GBB,10H      BBB,
107*      *4H LTA,4H LTB)
108*      270 FORMAT(10X,I10,F10.1,F10.4,2F10.2,F10.1,F10.4,2I4)
109*      C
110*      C COMPUT AINV(*),QQMIN(*) AND QQMAX(*)
111*      C
112*      NA=0
113*      NAA=1
114*      DO 360 I=1,NB
115*      NAA=NAA+NA
116*      NA=NGEN(I)
117*      IF(NA.EQ.0) GO TO 350
118*      NN=NAA-1
119*      QQMIN(I)=0.0

```

```

120*      QQMAX(I)=0.0
121*      AINV(I)=0.0
122*      DO 310 J=1,NA
123*      QQMIN(I)=QQMIN(I)+QMIN(NN+J)
124*      QQMAX(I)=QQMAX(I)+QMAX(NN+J)
125*      310 AINV(I)=AINV(I)+1.0/A2(NN+J)
126*      GO TO 360
127*      350 QQMIN(I)=0.0
128*      QQMAX(I)=0.0
129*      AINV(I)=0.0
130*      360 CONTINUE
131*      C
132*      C
133*      C      *****
134*      C      * MINIMIZE *
135*      C      *****
136*      C
137*      C      COMPUTE AN INITIAL GUESS
138*      C
139*      C      CALL CINITG(DCDP,COST,JFAIL)
140*      C      IF(JFAIL.NE.0) GO TO 1100
141*      C
142*      C      ESTIMATE MULTIPLIERS
143*      C
144*      C      DO 370 I=1,NB
145*      C      PLA(I)=DCDP
146*      C      QLA(I)=0.0
147*      C      QMY(I)=0.0
148*      C      370 VMY(I)=0.0
149*      C      DO 380 I=1,NG
150*      C      380 PMY(I)=0.0
151*      C
152*      C      MINIMIZE WITH FIXED MULTIPLIERS
153*      C
154*      C      N=2*NB
155*      C      MODE=1
156*      C
157*      C      DFN=10.0
158*      C      DO 390 I=1,N
159*      C      390 EPSX(I)=0.0001
160*      C      400 DO 410 I=1,NB
161*      C      X(I)=REAL(VB(I))
162*      C      410 X(I+NB)=AIMAG(VB(I))
163*      C
164*      C      415 CALL FLET(FUNC,N,X,F,G,H,W,DFN,EPSX,MODE,200,1,IEXIT)
165*      C      MODE=3
166*      C      IF(IEXIT.EQ.3) GO TO 415
167*      C      DFN=1.0
168*      C      IF(IEXIT.NE.1) GO TO 1100
169*      C
170*      C      CHECK CONSTRAINTS
171*      C
172*      C      NA=0
173*      C      NAA=1
174*      C      DO 600 I=1,NB
175*      C      NAA=NAA+NA
176*      C      NA=NGEN(I)
177*      C      IF(NA.EQ.0) GO TO 500
178*      C
179*      C      CHECK ACTIVE POWER LIMITS
180*      C
181*      C      NN=NAA-1
182*      C      DO 450 J=1,NA
183*      C      IF((PGEN(NN+J)-PMAX(NN+J)).GT.EPSPG) GO TO 620

```

```

184*      450 IF((PMIN(NN+J)-PGEN(NN+J)).GT.EPSP6) GO TO 640
185*      C
186*      C      CHECK REACTIVE POWER LIMITS
187*      C
188*      R=Q(I)+QD(I)
189*      IF((R-QQMAX(I)).GT.EPSQG) GO TO 660
190*      IF((QQMIN(I)-R).GT.EPSQG) GO TO 680
191*      GO TO 550
192*      C
193*      C      NO GENERATOR
194*      C      CHECK EQUALITY CONSTRAINTS
195*      C
196*      500 AP=ABS(P(I)+PD(I))
197*      IF(AP.GT.EPSP) GO TO 700
198*      IF(ABS(Q(I)+QD(I)).GT.EPSQ) GO TO 720
199*      C
200*      C      CHECK VOLTAGE LIMITS
201*      C
202*      550 IF((VABS(I)-VMAX(I)).GT.EPSV) GO TO 740
203*      IF((VMIN(I)-VABS(I)).GT.EPSV) GO TO 760
204*      600 CONTINUE
205*      C
206*      C      COMES HERE IF THE CONSTRAINTS SATISFIED
207*      C
208*      JMP=1
209*      GO TO 900
210*      C
211*      620 NNJ=NN+J
212*      WRITE(LP,630) NNJ
213*      630 FORMAT(/10X,31HPMAX LIMIT VIOLATED AT GEN.NR =,I5)
214*      GO TO 800
215*      640 NNJ=NN+J
216*      WRITE(LP,650) NNJ
217*      650 FORMAT(/10X,31HPMIN LIMIT VIOLATED AT GEN.NR =,I5)
218*      GO TO 800
219*      660 WRITE(LP,670) I
220*      670 FORMAT(/10X,33HQQMAX LIMIT VIOLATED AT BUS NR =,I5)
221*      GO TO 800
222*      680 WRITE(LP,690) I
223*      690 FORMAT(/10X,33HQQMIN LIMIT VIOLATED AT BUS NR =,I5)
224*      GO TO 800
225*      700 WRITE(LP,710) I
226*      WRITE(LP,705) AP
227*      705 FORMAT(/10X,3HAP=,F10.3)
228*      710 FORMAT(/10X,47HP EQUALITY CONSTRAINT NOT SATISFIED AT BUS NR =,I5)
229*      GO TO 800
230*      720 WRITE(LP,730) I
231*      730 FORMAT(/10X,47HQ EQUALITY CONSTRAINT NOT SATISFIED AT BUS NR =,I5)
232*      GO TO 800
233*      740 WRITE(LP,750) I
234*      750 FORMAT(/10X,32HVMAX LIMIT VIOLATED AT BUS NR =,I5)
235*      GO TO 800
236*      760 WRITE(LP,770) I
237*      770 FORMAT(/10X,32HVMIN LIMIT VIOLATED AT BUS NR =,I5)
238*      GO TO 800
239*      C
240*      C      ADJUST MULTIPLIERS
241*      C
242*      800 CALL ADJMP
243*      JMP=0
244*      C
245*      C
246*      C      *****
247*      C      * PRINT RESULT *

```

```

248*      C      *****
249*      900 WRITE(LP,910)
250*      910 FORMAT(///10X,42HRESULT FROM ECONOMIC LOAD DISPATCH PROGRAM/
251*      *10X,42(1H*))
252*      WRITE(LP,920)
253*      920 FORMAT(///10X,10H      BUS,12H      PINJ,12H      QINJ,
254*      *12H      VMIN,12H      VABS,12H      VMAX,12H      VANG)
255*      DO 930 I=1,NB
256*      930 WRITE(LP,940) I,PINJ(I),QINJ(I),VMIN(I),VABS(I),VMAX(I),VANG(I)
257*      940 FORMAT(10X,I10,6F12.4)
258*      WRITE(LP,950)
259*      950 FORMAT(///10X,10H      GEN.NR,12H      PMIN,
260*      *12H      PGEN,12H      PMAX)
261*      DO 960 I=1,NG
262*      960 WRITE(LP,970) I,PMIN(I),PGEN(I),PMAX(I)
263*      970 FORMAT(10X,I10,3F12.4)
264*      975 WRITE(LP,980)
265*      980 FORMAT(///10X,11HMULTIPLIERS/10X,10H      BUS,
266*      *10H      PLA,10H      QLA,10H      IQV,10H      QMY,
267*      *10H      IVV,10H      VMY)
268*      DO 990 I=1,NB
269*      990 WRITE(LP,1000) I,PLA(I),QLA(I),IQV(I),QMY(I),IVV(I),VMY(I)
270*      1000 FORMAT(10X,I10,2F10.4,I10,F10.4,I10,F10.4)
271*      WRITE(LP,1010)
272*      1010 FORMAT(//10X,10H      GEN.NR,10H      IPV,10H      PMY)
273*      DO 1020 I=1,NG
274*      1020 WRITE(LP,1030) I,IPV(I),PMY(I)
275*      1030 FORMAT(10X,2I10,F10.4)
276*      C
277*      WRITE(LP,1040) F,FO
278*      1040 FORMAT(///10X,43HGENERATOR COST FUNCTION INKL. PENALTIES =,
279*      *F10.3/10X,43HGENERATOR COST FUNCTION EXKL. PENALTIES =,F10.3/)
280*      IF(JMP.EQ,0) GO TO 400
281*      C
282*      1100 STOP
283*      END

```

END OF COMPILATION: NO DIAGNOSTICS.

# SUBROUTINE CINITG

```

1*      SUBROUTINE CINITG(DCDP,COST,JFAIL)
2*      C
3*      PARAMETER MB=10, MG=14, ML=13, MBB=20
4*      C
5*      COMPUTES AN INITIAL GUESS TO THE ECONOMIC LOAD DISPATCH PROGRAM
6*      C
7*      AUTHOR, JAHN RAGNARSSON 9-4-1973
8*      C
9*      DIMENSION
10*     *PMIN(MG), PMAX(MG), QQMIN(MB), QQMAX(MB), VMIN(MB), VMAX(MB),
11*     *GAA(ML), BAA(ML), RAB(ML), XAB(ML), GBB(ML), BBB(ML), LTA(ML), LTB(ML),
12*     *A1(MG), A2(MG), PGEN(MG), NGEN(MB), AINV(MB), NTYP(MB),
13*     *P(MB), PINJ(MB), PD(MB), Q(MB), QINJ(MB), QD(MB), VABS(MB), VANG(MB),
14*     *VB(MB),
15*     *Y(MB,MB), IB(MB)
16*     COMMON/BL 1/ NB,NG,NL,IPRINT

```

```

17*      COMMON/BL 2/ PMIN,PMAX,QQMIN,QQMAX,VMIN,VMAX
18*      COMMON/BL 4/ GAA,BAA,RAB,XAB,GBB,BBB,LTA,LTB
19*      COMMON/BL 6/ A1,A2,PGEN,NGEN,AINV,NTYP
20*      COMMON/BL 7/ P,PINJ,PD,Q,QINJ,QD,VABS,VANG,VB
21*      COMMON/BL10/ Y,IB
22*      COMPLEX VB,IB,Y,SL,NOLL
23*      DATA NOLL/(0.0,0.0)/,JJAC/1/,MAXIT/20/,EPS/1.0E-4/,EPZ/1.0E-4/
24*      C
25*      C
26*      C      DISPATCH ACTIVE POWER NEGLECTING TRANSMISSION LOSSES
27*      C
28*      PDEM=0.0
29*      DO 10 K=1,NB
30*      10 PDEM=PDEM+PD(K)
31*      LIM=1
32*      CALL ELDNL(A1,A2,PMIN,PGEN,PMAX,PDEM,EPZ,COST,DCDP,NG,IPRINT,
33*      1LIM)
34*      C
35*      C      PERFORM A LOAD-FLOW CALCULATION WITH MAXIMUM VOLTAGES
36*      C      AT NODES WITH SUFFICIENT REACTIVE CAPACITY
37*      C
38*      IGO=0
39*      15 DO 20 K=1,NB
40*      NTYP(K)=2
41*      IF(NGEN(K).EQ.0) GO TO 18
42*      VANG(K)=0.0
43*      GO TO 19
44*      18 QQMIN(K)=-1000.0
45*      QQMAX(K)=1000.0
46*      NTYP(K)=1
47*      QINJ(K)=-QD(K)
48*      19 VABS(K)=VMAX(K)
49*      20 CONTINUE
50*      NTYP(1)=3
51*      25 NAA=0
52*      NAA=2
53*      DO 50 K=2,NB
54*      PINJ(K)=0.0
55*      NAA=NAA+NA
56*      NA=NGEN(K)
57*      IF(NA.EQ.0) GO TO 40
58*      NN=NAA-1
59*      DO 30 L=1,NA
60*      30 PINJ(K)=PINJ(K)+PGEN(NN+L)
61*      40 PINJ(K)=PINJ(K)-PD(K)
62*      50 CONTINUE
63*      CALL LFLOW(GAA,BAA,RAB,XAB,GBB,BBB,LTA,LTB,PINJ,QINJ,QQMIN,QQMAX,
64*      *VMIN,VMAX,VABS,VANG,NTYP,EPS,NB,NL,JJAC,MAXIT,IPRINT,JFAIL,PD,QD)
65*      IF(JFAIL.NE.0) GO TO 100
66*      DO 55 I=1,NB
67*      VB(I)=CMPLX(VABS(I)*COS(VANG(I)),VABS(I)*SIN(VANG(I)))
68*      P(I)=PINJ(I)
69*      55 Q(I)=QINJ(I)
70*      IF(IGO.EQ.1) GO TO 100
71*      C
72*      C      COMPUTE TRANSMISSION LOSSES AND REDISPATCH ACTIVE POWER
73*      C
74*      DO 80 K=1,NB
75*      SL=NOLL
76*      DO 70 J=1,NB
77*      70 SL=SL+Y(K,J)*VB(J)
78*      80 IB(K)=SL
79*      PLOSS=0.0
80*      DO 90 K=1,NB

```

```
81*      90 PLOSS=PLOSS+VB(K)*CONJG(IB(K))
82*      PDEM=PDEM+PLOSS
83*      C
84*      CALL ELDNL(A1,A2,PMIN,PGEN,PMAX,PDEM,EPZ,COST,DCDP,NG,IPRINT,
85*      1LIM)
86*      C
87*      C      PERFORM A NEW LOAD-FLOW CALCULATION WITH MAXIMUM VOLTAGES
88*      C
89*      IGO=1
90*      GO TO 15
91*      100 RETURN
92*      END
```

END OF COMPILATION: NO DIAGNOSTICS.



# SUBROUTINE FUNC

```

1*      SUBROUTINE FUNC(X,N,F,G,FXX,IFXX,ICON,IERR)
2*      PARAMETER MB=10,MG=14,MBB=20
3*      DIMENSION
4*      *PMIN(MG),PMAX(MG),QQMIN(MB),QQMAX(MB),VMIN(MB),VMAX(MB),
5*      *PLA(MB),QLA(MB),PMY(MG),QMY(MB),VMY(MB),IPV(MG),IQV(MB),IVV(MB),
6*      *DPDE(MB,MB),DPDF(MB,MB),DQDE(MB,MB),DQDF(MB,MB),
7*      *A1(MG),A2(MG),PGEN(MG),NGEN(MB),AINV(MB),NTYP(MB),
8*      *P(MB),PINJ(MB),PD(MB),Q(MB),QINJ(MB),QD(MB),VABS(MB),VANG(MB),
9*      *VB(MB),GO(MBB),
10*     *G(1),X(1),FXX(IFXX,1),
11*     *Y(MB,MB),IB(MB),
12*     *VBR(MB),VBI(MB)
13*     COMMON/BL 1/ NB,NG,NL,IPRINT
14*     COMMON/BL 2/ PMIN,PMAX,QQMIN,QQMAX,VMIN,VMAX
15*     COMMON/BL 3/ PLA,QLA,PMY,QMY,VMY,IPV,IQV,IVV,C,D
16*     COMMON/BL 5/ DPDE,DPDF,DQDE,DQDF
17*     COMMON/BL 6/ A1,A2,PGEN,NGEN,AINV,NTYP
18*     COMMON/BL 7/ P,PINJ,PD,Q,QINJ,QD,VABS,VANG,VB
19*     COMMON/BL10/ Y,IB
20*     COMMON/BL11/ FO,GO
21*
22*     COMPLEX VB,IB,SL,NOLL,Y
23*     DATA NOLL/(0.0,0.0)/,LP/6/,EPZ/1.0E-4/
24*
25*     C
26*     C
27*     C
28*     C
29*     C
30*     C
31*     C
32*     C
33*     C
34*     C
35*     C
36*     C
37*     C
38*     C
39*     C
40*     C
41*     C
42*     DO 10 I=1,NB
43*     VBR(I)=X(I)
44*     VBI(I)=X(I+NB)
45*     VABS(I)=SQRT(VBR(I)**2+VBI(I)**2)
46*     VANG(I)=ATAN2(VBI(I),VBR(I))
47*     10 VB(I)=CMPLX(VBR(I),VBI(I))
48*     C
49*     C
50*     C
51*     DO 30 I=1,NB
52*     SL=NOLL
53*     DO 20 J=1,NB
54*     20 SL=SL+Y(I,J)*VB(J)
55*     30 IB(I)=SL
56*     C

```

```

57*      C      PRINT OUT?
58*      C
59*          IF(IPRINT) 40,40,60
60*      40 WRITE(LP,45)
61*      45 FORMAT (1H1,9X,30H          BUS          CURRENTS/10X,30(1H-))
62*          WRITE(LP,47)
63*      47 FORMAT(//17X,3HBUS,12H     REAL(IB),12H     AIMAG(IB))
64*          DO 50 I=1,NB
65*      50 WRITE(LP,55) I,IB(I)
66*      55 FORMAT(/10X,I10,2F12.4)
67*      C
68*      C      COMPUTE ACTIVE AND REACTIVE POWER
69*      C
70*      60 DO 70 I=1,NB
71*          P(I)=REAL(VB(I)*CONJG(IB(I)))
72*          PINJ(I)=P(I)
73*          Q(I)=AIMAG(VB(I)*CONJG(IB(I)))
74*      70 QINJ(I)=Q(I)
75*      C
76*      C      PRINT OUT?
77*      C
78*          IF (IPRINT=1) 80,80,120
79*      80 WRITE(LP,90)
80*      90 FORMAT(///10X,21HBUS VOLTAGE AND POWER/10X,21(1H-)//10X
81*          1,10H          BUS
82*          2,12H          VMIN
83*          3,12H          VABS
84*          4,12H          VMAX
85*          5,12H          VBR(I)
86*          6,12H          VBI(I)
87*          7,12H          P(I)
88*          8,12H          Q(I)
89*          DO 100 I=1,NB
90*          ABSO=CABS(VB(I))
91*      100 WRITE(LP,110) I,VMIN(I),ABSO,VMAX(I),VBR(I),
92*          1VBI(I),P(I),Q(I)
93*      110 FORMAT(10X,I10,7F12.4)
94*      C
95*      C      COMPUTE DPDE,DPDF,DQDE AND DQDF IF VB=E+JF
96*      C
97*      120 DO 140 I=1,NB
98*          DO 130 J=1,NB
99*          SL=VB(I)*CONJG(Y(I,J))
100*      DPDE(I,J)=REAL(SL)
101*      IF(I.EQ.J) DPDE(I,J)=DPDE(I,J)+REAL(IB(I))
102*      DPDF(I,J)=AIMAG(SL)
103*      IF(I.EQ.J) DPDF(I,J)=DPDF(I,J)+AIMAG(IB(I))
104*      DQDE(I,J)=AIMAG(SL)
105*      IF(I.EQ.J) DQDE(I,J)=DQDE(I,J)-AIMAG(IB(I))
106*      DQDF(I,J)=-REAL(SL)
107*      130 IF(I.EQ.J) DQDF(I,J)=DQDF(I,J)+REAL(IB(I))
108*      140 CONTINUE
109*      C
110*      C      PRINT OUT?
111*      C
112*          IF(IPRINT) 150,150,200
113*      150 WRITE(LP,160)
114*      160 FORMAT(//10X,4HDPE/10X,4(1H-)/)
115*          CALL MAPRI(DPDE,NB,NB,MB)
116*          WRITE(LP,170)
117*      170 FORMAT(///10X,4HDPDF/10X,4(1H-)/)
118*          CALL MAPRI(DPDF,NB,NB,MB)
119*          WRITE(LP,180)
120*      180 FORMAT(///10X,4HDQDE/10X,4(1H-)/)

```

```

121*      CALL MAPRI(DQDE,NB,NB,MB)
122*      WRITE(LP,190)
123*      190 FORMAT(///10X,4HDQDF/10X,4(1H-)/)
124*      CALL MAPRI(DQDF,NB,NB,MB)
125*      C
126*      C      COMPUTE F AND G(*)
127*      C
128*      200 F=0.0
129*          F0=0.0
130*          DO 210 I=1,NB
131*              G0(I)=0.0
132*              G0(I+NB)=0.0
133*              G(I)=0.0
134*          210 G(I+NB)=0.0
135*              NA=0
136*              NAA=1
137*          220 DO 600 I=1,NB
138*              NAA=NAA+NA
139*              NA=NGEN(I)
140*              IF(NA.EQ.0) GO TO 400
141*              PDEM=P(I)+PD(I)
142*              LIM=1
143*              IPRINT=IPRINT+1
144*              CALL ELDNL(A1(NAA),A2(NAA),PMIN(NAA),PGEN(NAA),PMA(NAA),
145*          1PDEM,EPZ,COST,DCDP,NA,IPRINT,LIM)
146*              IPRINT=IPRINT-1
147*              PLA(I)=DCDP
148*              F0=F0+COST
149*              DO 230 J=1,NB
150*                  G(J)=G(J)+DCDP*DPDE(I,J)
151*                  G(J+NB)=G(J+NB)+DCDP*DPDF(I,J)
152*                  G0(J)=G(J)
153*          230 G0(J+NB)=G(J+NB)
154*      C
155*      C      CHECK INEQUALITY CONSTRAINTS
156*      C
157*          NN=NAA-1
158*          PPMIN=0.0
159*          PPMAX=0.0
160*          DO 240 J=1,NA
161*              PPMIN=PPMIN+PMIN(NN+J)
162*          240 PPMAX=PPMAX+PMA(NN+J)
163*      C
164*      C      CHECK INEQUALITY CONSTRAINTS
165*      C
166*          DO 300 J=1,NA
167*              HP=PGEN(NN+J)-PMA(NN+J)
168*              IF(HP) 250,255,260
169*          250 HP=PMIN(NN+J)-PGEN(NN+J)
170*              IF(HP) 255,255,270
171*          255 IF(IPV(NN+J)) 270,300,260
172*      C
173*      C      ACTIVE POWER MAX LIMIT VIOLATED
174*      C
175*          260 T=PMA(NN+J)/PPMAX
176*              IF(IPV(NN+J)) 280,280,290
177*      C
178*      C      ACTIVE POWER MIN LIMIT VIOLATED
179*      C
180*          270 T=-PMIN(NN+J)/PPMIN
181*              IF(IPV(NN+J)) 290,280,280
182*      C
183*      C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
184*      C

```

```

185*      280 F=F+D*HP**3
186*          DO 285 K=1,NB
187*          G(K)=G(K)+3.0*D*T*DPDE(I,K)*HP**2
188*      285 G(K+NB)=G(K+NB)+3.0*D*T*DPDF(I,K)*HP**2
189*          GO TO 300
190*      C
191*      C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
192*      C
193*      290 F=F+PMY(NN+J)*HP+D*HP**2
194*          DO 295 K=1,NB
195*          G(K)=G(K)+PMY(NN+J)*T*DPDE(I,K)+
196*          12.0*D*T*DPDE(I,K)*HP
197*      295 G(K+NB)=G(K+NB)+PMY(NN+J)*T*DPDF(I,K)+
198*          *2.0*D*T*DPDF(I,K)*HP
199*          DO 298 K=1,NB
200*          GO(K)=GO(K)+PMY(NN+J)*T*DPDE(I,K)+2.0*D*T*DPDE(I,K)*HP
201*      298 GO(K+NB)=GO(K+NB)+PMY(NN+J)*T*DPDF(I,K)+2.0*D*T*DPDF(I,K)*HP
202*      300 CONTINUE
203*          R=Q(I)+QD(I)
204*          IF(R-QQMAX(I)) 305,307,310
205*      305 IF(QQMIN(I)-R) 307,307,320
206*      307 IF(IQV(I)) 320,500,310
207*      C
208*      C      REACTIVE POWER MAX LIMIT VIOLATED
209*      C
210*      310 HQ=R-QQMAX(I)
211*          T=1.0
212*          IF(IQV(I)-0) 330,330,340
213*      C
214*      C      REACTIVE POWER MIN LIMIT VIOLATED
215*      C
216*      320 HQ=QQMIN(I)-R
217*          T=-1.0
218*          IF(IQV(I)) 340,330,330
219*      C
220*      C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
221*      C
222*      330 F=F+D*HQ**3
223*          DO 335 J=1,NB
224*          G(J)=G(J)+T*DQDE(I,J)*3.0*D*HQ**2
225*      335 G(J+NB)=G(J+NB)+T*DQDF(I,J)*3.0*D*HQ**2
226*          GO TO 500
227*      C
228*      C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
229*      C
230*      340 F=F+QMY(I)*HQ+D*HQ**2
231*          DO 345 J=1,NB
232*          G(J)=G(J)+T*QMY(I)*DQDE(I,J)+T*D*2.0*HQ*DQDE(I,J)
233*      345 G(J+NB)=G(J+NB)+T*QMY(I)*DQDF(I,J)+T*D*2.0*HQ*DQDF(I,J)
234*          DO 350 J=1,NB
235*          GO(J)=GO(J)+T*QMY(I)*DQDE(I,J)+T*D*2.0*HQ*DQDE(I,J)
236*      350 GO(J+NB)=GO(J+NB)+T*QMY(I)*DQDF(I,J)+T*D*2.0*HQ*DQDF(I,J)
237*          GO TO 500
238*      C
239*      C      BUS WITHOUT GENERATOR
240*      C
241*      400 S=P(I)+PD(I)
242*          IF(S) 405,450,405
243*      405 F=F+PLA(I)*S+C*S**2
244*          DO 410 J=1,NB
245*          G(J)=G(J)+PLA(I)*DPDE(I,J)+2.0*C*S*DPDE(I,J)
246*      410 G(J+NB)=G(J+NB)+PLA(I)*DPDF(I,J)+2.0*C*S*DPDF(I,J)
247*          DO 420 J=1,NB
248*          GO(J)=GO(J)+PLA(I)*DPDE(I,J)+2.0*C*S*DPDE(I,J)

```

```

249*      420 G0(J+NB)=G0(J+NB)+PLA(I)*DPDF(I,J)+2.0*C*S*DPDF(I,J)
250*      450 R=Q(I)+QD(I)
251*          IF(R) 455,500,455
252*      455 F=F+QLA(I)*R+C*R**2
253*          DO 460 J=1,NB
254*          G(J)=G(J)+QLA(I)*DQDE(I,J)+2.0*C*R*DQDE(I,J)
255*      460 G(J+NB)=G(J+NB)+QLA(I)*DQDF(I,J)+2.0*C*R*DQDF(I,J)
256*          DO 470 J=1,NB
257*          G0(J)=G0(J)+QLA(I)*DQDE(I,J)+2.0*C*R*DQDE(I,J)
258*      470 G0(J+NB)=G0(J+NB)+QLA(I)*DQDF(I,J)+2.0*C*R*DQDF(I,J)
259*      C
260*      C      CHECK VOLTAGE LIMITS
261*      C
262*      500 IF(VABS(I)-VMAX(I)) 505,507,510
263*      505 IF(VMIN(I)-VABS(I)) 507,507,520
264*      507 IF(IVV(I)) 520,600,510
265*      C
266*      C      VOLTAGE MAX LIMIT VIOLATED
267*      C
268*      510 HV=VABS(I)-VMAX(I)
269*          A=1.0/VABS(I)
270*          IF(IVV(I)-0) 530,530,540
271*      C
272*      C      VOLTAGE MIN LIMIT VIOLATED
273*      C
274*      520 HV=VMIN(I)-VABS(I)
275*          A=-1.0/VABS(I)
276*          IF(IVV(I)) 540,530,530
277*      C
278*      C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
279*      C
280*      530 F=F+D*HV**3
281*          DO 535 J=1,NB
282*          G(J)=G(J)+REAL(VB(I))*A*3.0*D*HV**2
283*      535 G(J+NB)=G(J+NB)+AIMAG(VB(I))*A*3.0*D*HV**2
284*          GO TO 600
285*      C
286*      C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
287*      C
288*      540 F=F+VMY(I)*HV+D*HV**2
289*          AR=REAL(VB(I))*A
290*          AI=AIMAG(VB(I))*A
291*          DO 545 J=1,NB
292*          G(J)=G(J)+AR*VMY(I)+AR*D*2.0*HV
293*      545 G(J+NB)=G(J+NB)+AI*VMY(I)+AI*D*2.0*HV
294*          DO 550 J=1,NB
295*          G0(J)=G0(J)+AR*VMY(I)+AR*D*2.0*HV
296*      550 G0(J+NB)=G0(J+NB)+AI*VMY(I)+AI*D*2.0*HV
297*      600 CONTINUE
298*      610 F=F+F0
299*      C
300*      C      PRINT OUT?
301*      C
302*      620 IF(IPRINT-1) 630,630,1000
303*      630 WRITE(LP,640)
304*      640 FORMAT(///10X,46HREACTIVE POWER GENERATED AND VOLTAGE MAGNITUDE/
305*          *10X,46(1H-)//10X
306*          1,10H      BUS
307*          2,12H      QQMIN
308*          3,12H      Q+QD
309*          4,12H      QQMAX
310*          5,12H      VMIN
311*          6,12H      VABS
312*          7,12H      VMAX)

```

```

313*      DO 650 I=1,NB
314*      R=Q(I)+QD(I)
315*      650 WRITE(LP,660) I,QQMIN(I),R,QQMAX(I),VMIN(I),VABS(I),VMAX(I)
316*      660 FORMAT(10X,I10,6F12.4)
317*      WRITE(LP,670)
318*      670 FORMAT(///10X,22HACTIVE POWER GENERATED/10X,22(1H-)//10X
319*      1,10H GENERATOR
320*      2,12H          PMIN
321*      3,12H          PGEN
322*      4,12H          PMAX)
323*      DO 680 I=1,NG
324*      680 WRITE(LP,690) I,PMIN(I),PGEN(I),PMAX(I)
325*      690 FORMAT(10X,I10,3F12.4)
326*      WRITE(LP,700) F,F0
327*      700 FORMAT(///10X,43HGENERATOR COST FUNCTION INKL. PENALTIES =,
328*      1F10.3/10X,43HGENERATOR COST FUNCTION EXKL.PENALTIES =,F10.3/)
329*      IF(IPRINT) 705,705,740
330*      705 WRITE(LP,710)
331*      710 FORMAT(///10X,10H          BUS,10H          G(I),10H          G(I+NB))
332*      DO 720 I=1,NB
333*      720 WRITE(LP,730) I,G(I),G(I+NB)
334*      730 FORMAT(10X,I10,2F10.3)
335*      C
336*      C      PRINT OUT INPUT DATA?
337*      C
338*      740 IF(IPRINT-1) 750,1000,1000
339*      750 WRITE(LP,760)
340*      760 FORMAT(///10X,11HMULTIPLIERS/10X,11(1H-)/10X
341*      1,10H          BUS
342*      2,10H          PLA
343*      3,10H          QLA
344*      4,10H          QMY
345*      5,10H          VMY)
346*      DO 770 I=1,NB
347*      770 WRITE(LP,780) I,PLA(I),QLA(I),QMY(I),VMY(I)
348*      780 FORMAT(10X,I10,4F10.3)
349*      WRITE(LP,790)
350*      790 FORMAT(///10X,11HMULTIPLIERS/10X,11(1H-)/10X
351*      1,10H GENERATOR
352*      2,10H          PMY)
353*      DO 800 I=1,NG
354*      800 WRITE(LP,810) I,PMY(I)
355*      810 FORMAT(10X,I10,F10.3)
356*      WRITE(LP,820) C,D
357*      820 FORMAT(///10X,43HCOEFFICIENTS IN THE GENERATOR COST FUNCTION/
358*      110X,43(1H-)/10X
359*      2,2HC=,F10.3/
360*      310X,2HD=,F10.3)
361*      WRITE(LP,830)
362*      830 FORMAT(///10X,12HY-BUS MATRIX/10X,12(1H-))
363*      CALL CMAPRI(Y,NB,NB,MB)
364*      WRITE(LP,840)
365*      840 FORMAT(///10X,
366*      149HACTIVE(=1) OR INACTIVE(=0) INEQUALITY CONSTRAINTS/
367*      110X,49(1H-)/10X,10H          BUS
368*      2,10H          IQV
369*      3,10H          IVV)
370*      DO 850 I=1,NB
371*      850 WRITE(LP,860) I,IQV(I),IVV(I)
372*      860 FORMAT(10X,3I10)
373*      WRITE(LP,865)
374*      865 FORMAT(///10X,10H GENERATOR,10H          IPV)
375*      DO 870 I=1,NG
376*      870 WRITE(LP,880) I,IPV(I)

```

```
377*      880 FORMAT(10X,2I10)
378*          WRITE(LP,900)
379*      900 FORMAT(///10X,36HNUMBER GENERATORS AND DEMANDED POWER/
380*          110X,36(1H-)/10X
381*          2,10H      BUS
382*          3,10H      NGEN
383*          4,12H      PD
384*          5,12H      QD)
385*          DO 910 I=1,NB
386*      910 WRITE(LP,920) I,NGEN(I),PD(I),QD(I)
387*      920 FORMAT(10X,2I10,2F12.4)
388*      1000 IF(IPRINT-2) 1010,1010,1020
389*      1010 WRITE(LP,700) F,F0
390*      1020 RETURN
391*          END
```

END OF COMPILATION: NO DIAGNOSTICS.

---

# SUBROUTINE ADJMPL

```
1*      SUBROUTINE ADJMPL
2*      PARAMETER MB=10, MG=14, MBB=20
3*      C
4*      C      ADJUST MULTIPLIERS USED IN THE ECONOMIC LOAD DISPATCH PROGRAM
5*      C
6*      C      AUTHOR, JAHN RAGNARSSON 9-4-1973
7*      C
8*      DIMENSION
9*      *PMIN(MG), PMAX(MG), QQMIN(MB), QQMAX(MB), VMIN(MB), VMAX(MB),
10*     *PLA(MB), QLA(MB), PMY(MG), QMY(MB), VMY(MB), IPV(MG), IQV(MB), IVV(MB),
```



```

11*      *DPDE(MB,MB),DPDF(MB,MB),DQDE(MB,MB),DQDF(MB,MB),
12*      *A1(MG),A2(MG),PGEN(MG),NGEN(MB),AINV(MB),NTYP(MB),
13*      *P(MB),PINJ(MB),PD(MB),Q(MB),QINJ(MB),QD(MB),VABS(MB),VANG(MB),
14*      *VB(MB),
15*      *G(MBB),GO(MBB),
16*      *X(MBB),
17*      *Y(MB,MB),IB(MB)
18*      COMMON/BL 1/ NB,NG,NL,IPRINT
19*      COMMON/BL 2/ PMIN,PMAX,QQMIN,QQMAX,VMIN,VMAX
20*      COMMON/BL 3/ PLA,QLA,PMY,QMY,VMY,IPV,IQV,IVV,C,D
21*      COMMON/BL 5/ DPDE,DPDF,DQDE,DQDF
22*      COMMON/BL 6/ A1,A2,PGEN,NGEN,AINV,NTYP
23*      COMMON/BL 7/ P,PINJ,PD,Q,QINJ,QD,VABS,VANG,VB
24*      COMMON/BL 8/ G
25*      COMMON/BL 9/ X
26*      COMMON/BL10/ Y,IB
27*      COMMON/BL11/ FO,GO
28*      COMPLEX VB
29*      DATA LP/6/
30*      C
31*      C      PRINT OUT?
32*      C
33*      IF(IPRINT-2) 5,5,9
34*      5 WRITE(LP,6)
35*      6 FORMAT(1H1,9X,18HADJUST MULTIPLIERS/10X,18(1H-)/10X,10H
36*      *,10H      GEN.NR,7H      FXHX)
37*      9 NA=0
38*      NAA=1
39*      DO 400 I=1,NB
40*      NAA=NAA+NA
41*      NA=NGEN(I)
42*      IF(NA.EQ.0) GO TO 205
43*      C
44*      NN=NAA-1
45*      PPMIN=0.0
46*      PPMAX=0.0
47*      DO 10 J=1,NA
48*      PPMIN=PPMIN+PMIN(NN+J)
49*      10 PPMAX=PPMAX+PMAX(NN+J)
50*      C
51*      C      CHECK ACTIVE POWER LIMITS
52*      C
53*      IPN=0
54*      DO 100 J=1,NA
55*      HP=PGEN(NN+J)-PMAX(NN+J)
56*      IF(HP) 15,15,20
57*      15 HP=PMIN(NN+J)-PGEN(NN+J)
58*      IF(HP) 17,17,25
59*      17 IF(IPV(NN+J)) 25,100,20
60*      C
61*      C      ACTIVE POWER MAX LIMIT VIOLATED
62*      C
63*      20 T=PMAX(NN+J)/PPMAX
64*      IPN=1
65*      IF(IPV(NN+J)) 60,60,65
66*      C
67*      C      ACTIVE POWER MIN LIMIT VIOLATED
68*      C
69*      25 T=-PMIN(NN+J)/PPMIN
70*      IPN=-1
71*      IF(IPV(NN+J)) 65,60,60
72*      C
73*      C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
74*      C

```

BUS

```

75*      60 PMY(NN+J)=3.0*D*HP**2
76*      IPV(NN+J)=IPN
77*      GO TO 100
78*      C
79*      C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
80*      C
81*      65 PMY(NN+J)=PMY(NN+J)+2.0*D*HP
82*      IF(PMY(NN+J)) 70,70,100
83*      C
84*      C      TEST IF MULTIPLIER POSITIVE
85*      C
86*      70 PMY(NN+J)=0.0
87*      IPV(NN+J)=0
88*      GO TO 100
89*      C
90*      100 CONTINUE
91*      C
92*      C      CHECK REACTIVE POWER LIMITS
93*      C
94*      110 R=Q(I)+QD(I)
95*          IQN=0
96*          IF(R-QQMAX(I)) 115,115,120
97*      115 IF(QQMIN(I)-R) 117,117,130
98*      117 IF(IQV(I)) 130,310,120
99*      C
100*     C      REACTIVE POWER MAX LIMIT VIOLATED
101*     C
102*     120 HQ=R-QQMAX(I)
103*         T=1.0
104*         IQN=1
105*         IF(IQV(I)) 170,170,180
106*     C
107*     C      REACTIVE POWER MIN LIMIT VIOLATED
108*     C
109*     130 HQ=QQMIN(I)-R
110*         T=-1.0
111*         IQN=-1
112*         IF(IQV(I)) 180,170,170
113*     C
114*     C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
115*     C
116*     170 QMY(I)=3.0*D*HQ**2
117*         IQV(I)=IQN
118*         GO TO 310
119*     C
120*     C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
121*     C
122*     180 QMY(I)=QMY(I)+2.0*D*HQ
123*     C
124*     C      TEST IF MULTIPLIER POSITIVE
125*     C
126*         IF(QMY(I)) 190,190,310
127*     190 QMY(I)=0.0
128*         IQV(I)=0
129*         GO TO 310
130*     C
131*     C      CHECK EQUALITY CONSTRAINTS
132*     C
133*     205 S=P(I)+PD(I)
134*         IF(S) 210,220,210
135*     210 PLA(I)=PLA(I)+2.0*C*S
136*     220 R=Q(I)+QD(I)
137*         IF(R) 230,310,230
138*     230 QLA(I)=QLA(I)+2.0*C*R

```

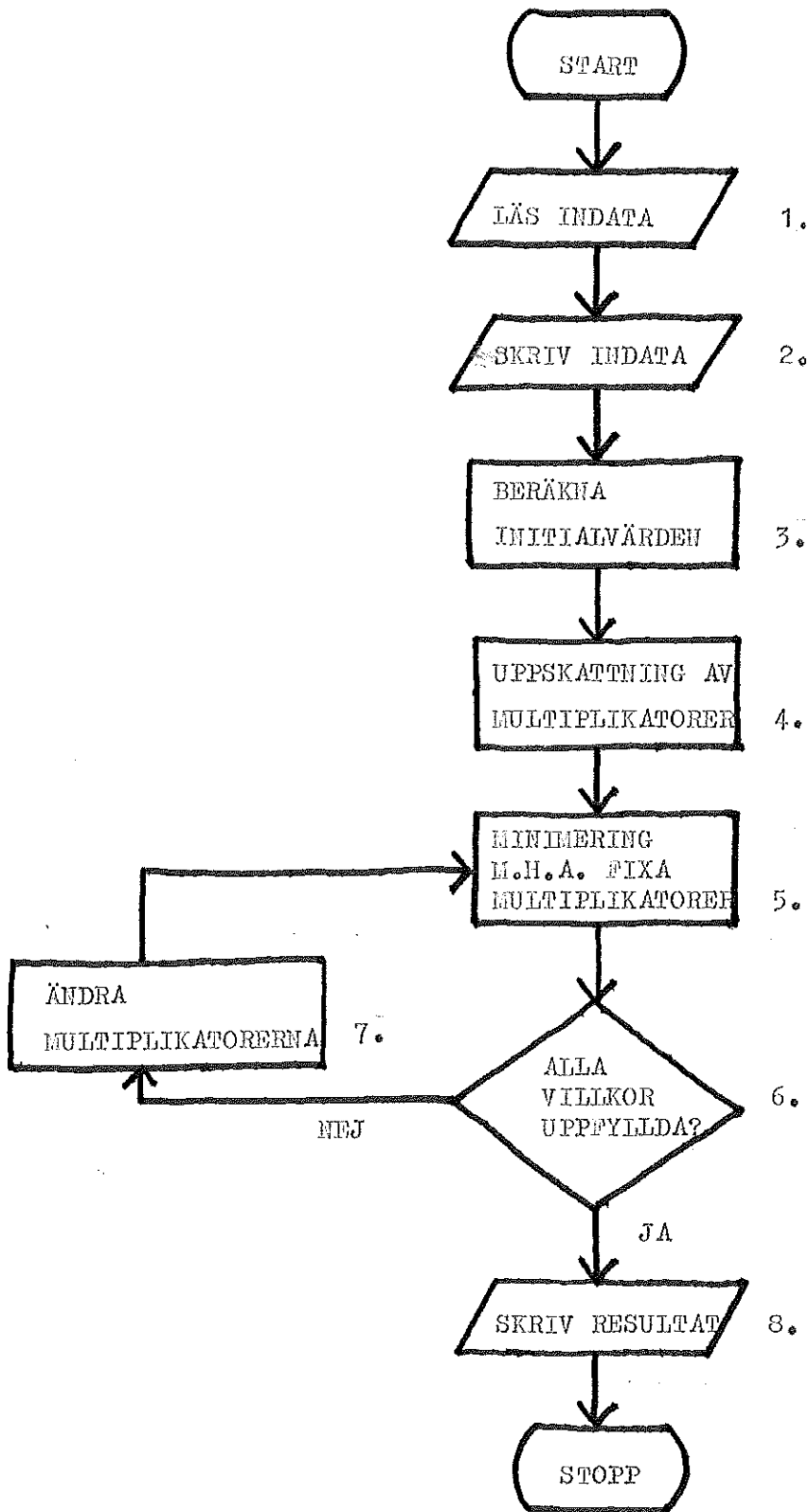
```

139*      C
140*      C      CHEK VOLTAGE LIMITS
141*      C
142*      C
143*      310  IVN=0
144*          IF(VABS(I)-VMAX(I)) 315,315,320
145*      315  IF(VMIN(I)-VABS(I)) 317,317,330
146*      317  IF(IVV(I)) 330,400,320
147*      C
148*      C      VOLTAGE MAX LIMIT VIOLATED
149*      C
150*      320  HV=VABS(I)-VMAX(I)
151*          A=1.0/VABS(I)
152*          IVN=1
153*          IF(IVV(I)) 370,370,380
154*      C
155*      C      VOLTAGE MIN LIMIT VIOLATED
156*      C
157*      330  HV=VABS(I)-VMIN(I)
158*          A=-1.0/VABS(I)
159*          IVN=-1
160*          IF(IVV(I)) 380,370,370
161*      C
162*      C      VIOLATED INACTIVE INEQUALITY CONSTRAINT
163*      C
164*      370  VMY(I)=3.0*D*HV**2
165*          IVV(I)=IVN
166*          GO TO 400
167*      C
168*      C      VIOLATED ACTIVE INEQUALITY CONSTRAINT
169*      C
170*      380  VMY(I)=VMY(I)+2.0*D*HV
171*          IF(VMY(I)) 390,390,400
172*      C
173*      C      TEST IF MULTIPLIER POSITIVE
174*      C
175*      390  VMY(I)=0.0
176*          IVV(I)=0
177*          GO TO 400
178*      C
179*      C
180*      400  CONTINUE
181*          RETURN
182*          END

```

END OF COMPILATION: NO DIAGNOSTICS.

# " FLÖDESSCHEMA:



OBS!

Flödesschema på "baksidan" till  
sidorna 7, 8, 9 o 10