

효율적인 電力潮流計算 프로그래밍에 관한 研究

An Efficient Power Flow Programming

論 文
29-12-3

金 俊 鉉* · 黃 甲 珠**
(Joon-Hyun Kim · Kab-Ju Hwang)

Abstract

The specific objective of this paper is to develop an useful power flow program for real time applications. Great emphasis has been put to the combination of algorithms and programming techniques for high speed, low storage, reliability, versatility and simplicity. A practical application in several cases has produced satisfiable results.

1. 緒 論

電力潮流計算은 電力系統의 計劃, 運用, 解析등에서 기본이 되는 算法으로 目的과 方法에 따라 種類가 다양하다. 1956년부터 발표되기 시작한 潮流計算은 크게 나누어 Y行列 反復法^{1,2)}, Z行列 反復法³⁾, Newton(-Raphson)法^{4,5)} 등을 들 수 있으며, 최근에는 電力系統의 複雜化와 自動化추세에 따라 安全度(Security)까지도 고려한 강력한 潮流計算法^{6,8)}들이 발표되고 있다. 특히 自動給電시스템과 같은 비교적 小形 計算機로 Real time 처리 가능한 解를 얻기 위하여는 從前의 論文들이 提案한 數學的인 正式化는 물론 프로그래밍 자체가 문제로 된다⁹⁾. 여기서 制約된 計算時間, 記憶量, 計算機能등 여러 條件들은 서로 排他的이므로, 본 研究에서는 機能別 알고리즘을 선정하고 效果的인 프로그래밍 技法에 의해 組合함으로써 條件에 부합된 解를 얻도록 하는 데 있다. 이러한 目的達成을 위하여 다음 다섯 가지 要求條件을 前提하고 프로그래밍을 한 다음 그 效用性을 評價하였다.

- 가. 大規模 系統이 制御시스템에 의해 Real time 처리가 될 수 있을 충분한 高速性
- 나. 미니 컴퓨터급 小形計算機에서도 적용가능한 經濟性
- 다. 母線이나 線路脫落 등 異常條件에서도 收斂이

양호한 信賴性

라. 系統計劃, 運用的 重要한 Q-V調整, Contingency 解析 등의 柔軟性.

마. 系統變更에 따른 入力 Data의 코딩, 알고리즘의 改善, 프로그램의 修正 등의 容易性

2. 기본 알고리즘

임의의 k 母線에서 有效 및 無効電力의 偏差式은

$$\Delta P_k = P_k^{sp} - V_k \sum_{m \neq k} V_m (G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km}) \quad (1)$$

$$\Delta Q_k = Q_k^{sp} - V_k \sum_{m \neq k} V_m (G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km}) \quad (2)$$

여기서 P_k^{sp} , Q_k^{sp} = k 母線의 指定 有效 및 無効電力
 θ_k , V_k = k 母線의 電壓角 및 크기

$$\theta_{km} = \theta_k - \theta_m$$

$G_{km} + jB_{km}$ = 母線어드미턴스 行列의 (k, m)번 要素

汎用 潮流計算法으로 널리 사용되는 Newton(-Raphson)法⁵⁾은

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (3)$$

으로 Jacobian 行列式을 反復에 의해 ΔQ , ΔV 를 修正함으로써 解를 얻게 된다. 식 (3)의 Jacobian 行列 記憶에 따른 經濟性을 고려한 Decoupled Newton 法⁶⁾은

$$[\Delta P/V] = [H][\Delta \theta] \quad (4)$$

$$[\Delta Q/V] = [L][\Delta V] \quad (5)$$

윗식의 $[H]$, $[L]$ 계산과정에서 몇 가지 일반적인 가정 ($\cos \theta_{km} \approx 1$, $G_{km} \sin \theta_{km} \ll B_{km} \cos \theta_{km}$, $Q_k \ll B_{kk}$)

* 正會員 : 漢陽大 工大 電氣工學科 教授

** 正會員 : 漢陽大 大學院 電氣工學科

接受日字 : 1980年 11月 3日

V_k)을 하여 간략화한 Fast Decoupled法⁷⁾은 다음과 같이 된다.

$$[\Delta P/V] = [B'][\Delta \theta] \quad (6)$$

$$[\Delta Q/V] = [B''][\Delta V] \quad (7)$$

여기서 $B'_{km} = -1/X_{km}, k \neq m$

$$B'_{kk} = \sum_{k \neq m} 1/X_{km}, B''_{km} = -B_{km}$$

본 研究에서도 Fast Decoupled 法을 導入하였으며, 문제가 되는 $P-\theta, Q-V$ 修正過程은 매 反復시 ΔP_{max} 와 ΔQ_{max} 중 큰 것을 우선 修正하는 方法을 택하였다.⁸⁾ 특히 본 研究에서는 Fast Decoupling 가정이 不良한 惡條件에서의 收斂性을 높이기 위하여 i 反復 收斂過程에서 ΔP_i 修正시 ΔQ_i 가 ΔQ_{i-1} 보다 커질 때는 ΔV_i 대신 ΔV_{i-1} 을 사용하여 ΔQ_i 를 再修正하도록 함으로써 (ΔQ_i 수정시도 마찬가지) 비교적 發散하는 問題도 收斂이 가능하게 되어 信賴性을 향상시켰다.

3. 逆行例 알고리즘

식 (6)과 식 (7)의 行列 $[B']$ 와 $[B'']$ 는 系統이 커짐에 따라 記憶量이 현저하게 증가하기 때문에 小形計算機로는 처리할 수 없게 된다. 본 研究에서는 效果的인 Storage scheme에 의해 $[B']$ 와 $[B'']$ 중 非零要素만을 1차원 배열하여 前提한 經濟性을 유지시키고, 이에 따른 母線과 線路番號로 이루어지는 Indexing scheme에 의해 引用함으로써 高速性도 유지하도록 하였다. 그리고 V 와 θ 를 修正하기 위해서는 $[B']$ 와 $[B'']$ 의 逆行列을 구해야 하므로, 三角化 直接法¹⁰⁾을 導入하여 초기에 $[L][D][U]$ 分解를 해둔 다음, 매 反復시마다 前進→對角→後進의 過程에 의해 $\Delta \theta$ 와 ΔV 가 구해지도록 하였다. 프로그래밍시 이 過程을 식 (8)~(10)과 같이 분류하고 함으로써 유사한 變形式($Q-V$ Sensitivity 計算 및 Contingency Simulation 등)에서도 쉽게 引用될 수 있도록 하였다. 즉,

가. 前進消去(Forward Elimination)

$$[U]^T[Y] = [B], [Y] = [D][U][X] \quad (8)$$

나. 對角代入(Diagonal Substitution)

$$[D][Y]' = [Y], [Y]' = [U][X] \quad (9)$$

다. 後進代入(Backward Substitution)

$$[U][X] = [Y]' \quad (10)$$

4. 母線 Ordering 알고리즘

$[B']$ 와 $[B'']$ 는 매우 Sparse한 行列이나 三角化시 생기는 fill-in항에 의해 그 逆行列이 반드시 Sparse하지는 않으며, 이 때의 Sparsity가 前提한 高速性과 經

濟性에 關여하게 되므로 Ordering이 필요로 된다. 이에 關해서는 Graphic 理論에 입각하여 많은 論文이 발표된 바 있으며, Tinney 氏 등¹⁰⁾은 消去過程의 순서를 정하는 몇가지 Scheme을 提案하였고, Irisarri 氏 등¹¹⁾은 DP 法에 의한 最適 Ordering 알고리즘을 提案하였다.

본 研究에 의하면 DP 法이나 Tinney 氏의 Scheme-3이 理論上 最適이긴 하지만, 프로그래밍시 Scratch 領域이 필요하고 系統이 커지면 Ordering에 따른 전체적인 高速性이 鈍化될 수가 있으므로, Tinney 氏의 Scheme-2에 입각하여 먼저 Slack 母線을 취한 다음 消去母線과 가장 가깝고 가장 적은 線路를 가진 母線을 취하도록 프로그래밍하였다. 또한 이미 作成된 Indexing table을 이용함으로써 記憶量의 증가가 없이 쉽게 Ordering 되도록 하였으며 事例研究결과 충분히 만족되었다. 그림 2는 그림 1의 모델系統이 Ordering 되는 過程을 나타내는 것으로, 三角化시 생기는 fill-in 항은 14개에서 10개로 줄어든다. 또한 Ordering 된 母線番號와 원래 母線番號와의 相互引用을 위하여

NEW(Ordered 母線番號) = 원래 母線番號

OLD(원래 母線番號) = Ordered 母線番號

와 같이 Variable Scheme을 이용함으로써 프로그래밍의 容易性도 도모하였다.

5. 電壓·無効電力調整 알고리즘

$P-V$ 母線에서 指定된 電壓을 유지하기 위하여는 無効電力 上·下限值를 벗어나는 경우가 있으므로 運轉可能범위내로 電壓을 調整할 필요가 있다. 이때 上·下限值를 벗어나는 $P-V$ 母線을 $P-Q$ 母線으로 바꾸어 計算하고, 逆으로 $P-Q$ 母線의 電壓을 調整하기 위해서는 $P-V$ 母線으로 바꾸면 된다. 그러나 이 작업은 매우 간단하지만 $[B'']$ 를 다시 三角化 해야 하므로 高速性에 영향을 주게 된다. 본 研究에서는 $[B'']$ 로부터 $Q-V$ Sensitivity를 구하여 偏差軌還技法에 의하여 調整되도록 프로그래밍하였다. 즉 $P-V_k$ 母線에서 n 反復제 Q_k 가 上·下限值를 벗어 났다면

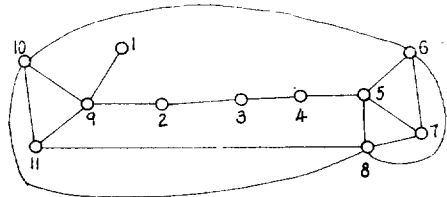


그림 1. 모델系統
Fig. 1. Model System

NO	1	2	3	4	5	6	7	8	9	10	11	
INDEX	1	3	6	9	12	17	22	26	32	37	42	46
KOL	1	2	3	4	5	6	7	8	10	11	D	
NEW												OLD
1	X	X							X			1
2	X	X	X	X	X				X	X		2
3	X	X	X	X	X	X	X		X	X		3
4	X	X	X	X	X	X	X	X	X	X		4
9	X	X	X	X	X	X	X	X	X	X	X	8
11	X	X	X	X	X	X	X	X	X	X	X	7
10	X	X	X	X	X	X	X	X	X	X	X	6
5	X	X	X	X	X	X	X	X	X	X	X	9
8	X	X	X	X	X	X	X	X	X	X	X	5
6	X	X	X	X	X	X	X	X	X	X	X	11
7	X	X	X	X	X	X	X	X	X	X	X	10

그림 2. 모델系統의 消去過程
Fig. 2. Bus Ordering Scheme of Model System

$$Q_k(V_k^n) \geq Q_k$$

$$V_k = V_k - S_k \frac{\Delta Q_k}{V_k} \tag{11}$$

식 (11)의 Sensitivity S_k 는 다음의 過程에 의하여 쉽게 計算된다.

$$\left(\frac{\Delta Q/V}{\Delta Q/VG_k} \right) = \left(\frac{B''}{C_1^T} \right) \left(\frac{C_1}{C_2} \right) \left(\frac{\Delta V}{\Delta VG_k} \right) \tag{12}$$

이 식에서 추가된 行列 C_1 은 해당 P-Q 母線의 $[B'']$ 값이다. 이 식을 分解하여

$$\frac{\Delta Q}{V} = [B''] \Delta V + C_1 \Delta VG_k \tag{13}$$

$$\frac{\Delta Q}{VG_k} = C_1^T \Delta V + C_2 \Delta VG_k \tag{14}$$

식 (13)을 三角化하면

$$\Delta V = [U]^T [D] [U] \left(-\frac{\Delta Q}{V} \right) = 0 = -[U]^T [D] [U]$$

$$C_1 \Delta VG_k$$

이 식을 식 (14)에 代入하고 $C_1 = [U]^T [U] C_1$ 으로 두면

$$\frac{\Delta Q}{VG_k} = [C_2 - C_1^T [D] C_1] \Delta VG_k$$

또는

$$\frac{\Delta Q}{VG_k} = \left[C_2 - \sum_{j=1}^{NB-1} \frac{(C_j)^2}{D_{jj}} \right] \Delta VG_k \tag{15}$$

여기서 Sensitivity S_k 는

$$S_k = \left[C_2 - \sum_{j=1}^{NB-1} \frac{(C_j)^2}{D_{jj}} \right]^{-1} \tag{16}$$

로 주어지

$$\Delta V_k = S_k \cdot \frac{\Delta Q_k}{V_k} \tag{17}$$

이 計算은 식 (7)을 처리할 때 간단히 함께 計算되므로 計算速度가 향상된다.

6. Contingency 알고리즘

自動給電시스템의 安全度기능을 위한 Contingency Simulation은 특히 計算速度가 문제가 된다. 본 研究에서는 Householder 公式¹²⁾을 導入하여 $[B'']$ 의 變形이 없이 計算되도록 프로그래밍하여 高速성을 높였다. 즉 식 (6)과 식 (7)에서 이용된 行列을 $[B_0]$, Contingency용 行列을 $[B_1]$ 이라 하면

$$[B_1] = [B_0] - b_1 \underline{M}_1^T \underline{M}_1 \tag{18}$$

여기서 $b = \frac{1}{X_L}$, X_L =脫落線路의 리액턴스

M =Branch-Node incidence 行列

$$[B_1]^{-1} = [B_0]^{-1} - C_1 \underline{X}_1 \underline{M}_1 [B_0]^{-1} \tag{19}$$

여기서 $C_1 = 1/b_1 + \underline{M}_1 \underline{X}_1$

식 (19)는 여러개의 線路脫落까지 확장시킬 수 있으며, 프로그래밍시 동시에 3개의 線路가 脫落되는 것까지 고려하였다. 이 過程은 다음과 같은 단계로 이루어진다.

가. Step 1 : $\underline{X}_1 = [B']^{-1} \underline{M}_1^T$, $C_1 = (1/b_1 + \underline{M}_1 \underline{X}_1)$

나. Step 2 : $\underline{X}_2 = [B']^{-1} \underline{M}_2^T$

$$\underline{X}_2 = \underline{X}_2 - C_1 \underline{X}_1 \underline{M}_1 \underline{X}_2, C_2 = (1/b_2 + \underline{M}_2 \underline{X}_2)$$

다. Step 3 : $\underline{X}_3 = [B']^{-1} \underline{M}_3^T$

$$\underline{X}_3 = \underline{X}_3 - C_1 \underline{X}_1 \underline{M}_1 \underline{X}_3$$

$$\underline{X}_3 = \underline{X}_3 - C_2 \underline{X}_2 \underline{M}_2 \underline{X}_3, C_3 = (1/b_3 + \underline{M}_3 \underline{X}_3)$$

라. Step 4 : $\Delta \theta_0 = [B']^{-1} \Delta P$

$$\begin{aligned} \Delta\theta_1 &= \Delta\theta_0 - C_1 X_1 M_1 \Delta\theta_0 \\ \Delta\theta_2 &= \Delta\theta_1 - C_2 X_2 M_2 \Delta\theta_1 \\ \Delta\theta_3 &= \Delta\theta_2 - C_3 X_3 M_3 \Delta\theta_2 \end{aligned}$$

ΔV 에 대해서도 $[B']$ 를 이용하여 위의 과정에 의해 구해지게 된다. 또한 發電機 脫落의 경우는 ΔP 와 ΔQ 를 변경함으로써 追加反復에 의해 구해 지도록 하였다.

7. 소프트웨어

본 研究에서는 前提한 條件에 부합되도록 프로그래밍시 그림 3과 같이 前述한 여러 技法들을 10개의 副 프로그램으로 블록화하여 프로그램간의 Interface를 원활하게 하였다. 主 프로그램은 線路와 母線 및 Contingency Data를 처리하고, 制御變數에 따라 副 프로그램의 처리과정을 制御한다. CHANGE 블록에서는 母線 및 線路데이터의 變更, 削除, 追加機能을 담당하며 INDEX 블록에서는 線路的 母線番號를 分類하여 Indexing table을 작성한다. 이 Indexing table을 이용하여 ORDER→YMAT→BFAC→SOLVE→FLOW의 과정으로 解가 구해진다. 프로그래밍시 引用이 잦은 $[Y]$ 는 非零要素 전부를 記憶시켜 Retrieval 時間을 감소시켰으며, $[B']$ 와 $[B'']$ 는 半만 記憶토록 하였다. 또한 制御시스템과의 Interface를 위하여 그림 4와 같

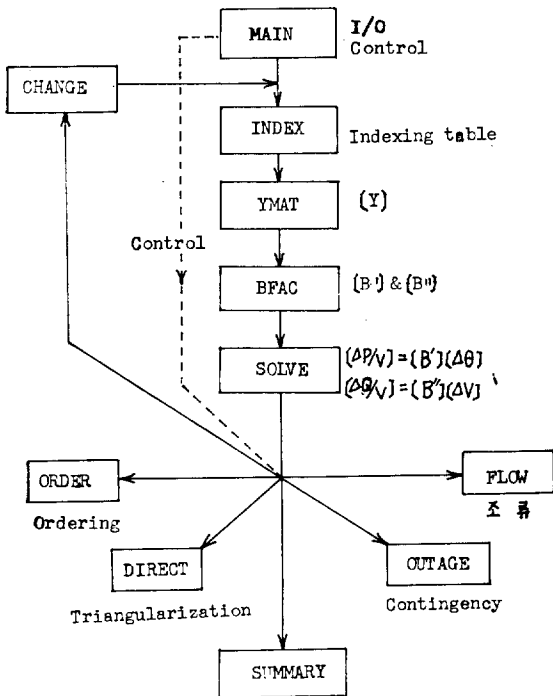


그림 3. 프로그램 구성도
Fig. 3. Program Flow

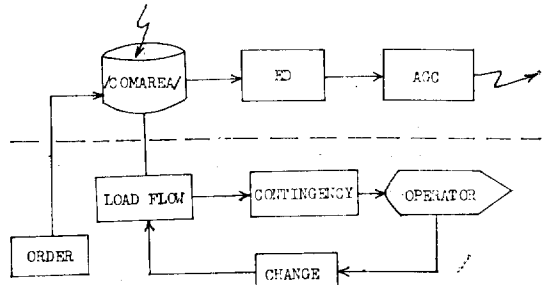


그림 4. 制御시스템과의 結合
Fig. 4. Interface for AGC/SCADA System

이 Real time 데이터의 共同領域(Comarea)을 引用토록 하므로써 CRT를 이용한 Simulation을 가능하게 하였다. 만약 본 프로그램이 制御시스템에 永住시키고자 할 때는 ORDER 블록에 의해 미리 Ordering을 한 다음 그림 4와 같이 구성하는 것이 바람직 할 것이다.

8. 事例研究

본 研究의 効用性 評價를 위하여 Random하게 母線番號가 주어진 16母線(韓電 京仁系統), 25母線(IEEE-25 系統), 48母線(韓電 '77系統), 82母線(韓電 '80系統)에서 각 블록별 諸量들을 實測한 主要결과를 要約하면 표 1과 같이 된다. 각 블록별 計算時間은 현재 韓國電力(株)에서 사용중인 FACOM-M180 大形計算機에서 얻어진 CPU 時間이며, 自動給電시스템의 CP-400에서 實測한 전체 計算時間은 82母線의 경우 3.4秒가

표 1. 事例研究 結果表
Table 1. Results for Actual Systems.

母線數	16	25	48	82	
線路數	18	35	93	127	
MW + j MVAR	1540 + j607	746 + j216	3769 + j 1235	4321 + j 2331	
반복횟수 $\epsilon < 10^{-3}$	4	4	4 1/2	5	
Sparsity (%)	79.7	84.8	89.8	95.4	
Fill-in항	8	21	50	82	
計算時間 (sum)	INDEX	2	6	25	48
	YMAT	2	4	13	14
	BFAC	4	10	34	84
	ORDER	2	3	7	15
	SOLVE	45	118	248	407
FLOW	64	119	316	399	
OUTAGE	10	28	48	75	
비고	韓電 '京仁	IEEE-25	韓電 '77	韓電 '80	

소요되었다. 프로그램을 최대 100母線, 200線路, 550非
需要素까지를 記憶토록 하였을 때 記憶量은 약 32KW
가 필요로 되었다. 표 2는 현재 韓電 中央給電指令所에
서 사용중인 82母線, 127線路의 收斂過程을 보여 주며
2회선 脫落의 Contingency에서 3회정도의 추가반복
이 필요로 되었다.

표 2. 韓電 '80系統의 收斂過程
Table 2. Convergence Process of KECO '80
System.

受復	ΔP_{max}	ΔQ_{max}	$\sum \Delta P_i $	$\sum \Delta Q_i $
1	731.8	720.1	7783	6242
2	-251.8	-111.8	1861	302.2
3	23.00	-12.31	86.97	18.95
4	3.665	-1.866	11.00	2.701
5	0.6003	-0.2853	1.543	0.4795

9. 結 論

본 研究는 電力系統運用 自動化를 위한 소프트웨어
개발의 일부로, 다섯가지 制約을 前提하고 이를 만족
시키기 위한 效果인 프로그래밍을 試圖하였다. 그
方法으로는 機能別 알고리즘을 선정하고, 프로그래밍
技法에 의해 組合하므로써 條件에 맞는 解를 얻도록
하였다. 이를 위한 주요 技法과 結果를 要約하면

가. Fast Decoupled 法²⁾을 導入하여 高速性を 向上
시켰고, 이때 Decoupling 假定이 成立되지 않는 惡條
件에서의 信賴性を 提高하였다.

나. Q-V Sensitivity 計算이나 Contingency 計算
시 Matrix技法에 의해 不變 三角行列을 이용하도록 하
므로써 高速性を 향상시켰다.

다. 1차원 配列을 위한 Indexing Scheme 을 이용
하여 記憶量의 증가가 없이 母線 Ordering 을 하고,
Variable Scheme에 의해 引用하므로써 經濟性도 높
였다.

라. 본 研究에서 導出한 프로그램으로 32KW 정도의
小形 計算機로 100母線 정도의 系統이 Real time처리
될 수 있음을 보였다.

끝으로 본 研究는 現代그룹의 教授研究獎勵基金으로
이루어진 것을 부언하며 謝意를 표한다.

參 考 文 獻

1. J.B. Ward, H.W. Hale; "Digital computer solution of power flow problems", AIEE Trans., Vol. PAS-75, June 1956.
2. A.F. Glimn, G.W. Stagg; "Automatic calculation of load flows", AIEE Trans., Vol. PAS-76, Oct. 1957.
3. P.P. Gupta, M.W. Humphrey Davies; "Digital computers in power system analysis", Proc. IEE, Vol. 108A, Jan. 1961.
4. J.E. Van Ness; "Iteration methods for digital load flow studies," AIEE Trans., Vol. PAS-78, Aug. 1959.
5. W.F. Tinney, C.E. Hart; "Power flow solution by Newton's method", IEEE Trans., Vol. PAS-86, Nov. 1967.
6. B. Stott; "Decoupled Newton load flow", IEEE Trans., Vol. PAS-91, Sept./Oct. 1972.
7. B. Stott, O. Alsac; "Fast decoupled load flow," IEEE Trans., Vol. PAS-93, May/June 1974.
8. 朴永文, 白榮植; 電力潮流計算의 改善에 관한 研究" 大韓電氣學會誌, Vol. 26, No. 2, 1977.
9. Ralph A Willoughby(Editor); "Sparse matrix proceedings" IBM Research, New York, Mar. 1969.
10. W.F. Tinney, M.S. Mayer; "Solution of large sparse system by ordered triangular factorization", IEEE Trans., Vol. AC-18, No. 4, Aug. 1973.
11. C. Irisarri, A.M. Sasson, S.F. Hodges; "An optimal ordering algorithm for sparse matrix applications", IEEE Trans., Vol. PAS-97, No. 6, Nov./Dec. 1978.
12. Householder, A.S.; "Principles of Numerical Analysis", McGraw-Hill Book Co., New York, P. 79, 1953.