

대규모 전력계통의 적용을 위한 전력조류계산법

이창근 · 황갑주
전기전자정보시스템공학부

<요 약>

전력조류계산은 전력계통의 상태를 파악하는데 필요한 도구로 전력계통의 운용과 계획의 기반이 되고 있다. 특히 대상 전력계통의 규모가 커지고 구성설비가 복잡해짐에 따라 보다 효과적으로 전력계통을 해석할 수 있는 조류계산법의 연구가 요청되고 있으며, 현재 우리나라 전력회사의 실무에서는 미국 PTI사에서 개발한 PSS/E 패키지를 수입하여 사용하고 있다. 본 연구는 실계통 규모를 해석할 수 있는 전력조류계산법을 개발하는데 있다. 개발된 산법에는 전력방정식을 풀기 위해 가우스-자이델법과 뉴턴-랩슨법 및 고속분할법을 포함하였으며, 스퍼스 프로그래밍 기법과 객체 지향적 프로그래밍 기법도 도입하였다. 개발한 산법의 성능을 평가하기 위해 IEEE 샘플계통과 우리나라 실계통에 적용하였으며, 그 결과를 실무에서 널리 사용하고 PSS/E의 결과와도 비교하였다.

A Power Flow Calculation Method for Large Scaled Power System Application

Lee, Chang-Gun · Hwang, Kab-Ju
Scool of Electrical Engineering

<Abstract>

Power flow calculation is a important tool for power system analysis, operation and planning. Therefore many methods to analysis power system have been proposed for decades. In Korea power system, the softwares are developed by the oversea and are not suitable for new computer environments. In this paper focused on developing software which can analyze large scaled power system and be suitable for new

computer environment. Developed software includes three well-known algorithms, that is, Gauss-Seidel, Newton-Raphson and Fast-Decoupled method. And it also employees a sparse programming technique, bus ordering scheme and object-oriented programming (OOP). The developed software is tested on IEEE -14, -30, -57 and -118 bus sample system and KEPCO's 823 bus system and their results are compared with those of PSS/E developed by PTI.

1. 서 론

전력계통의 각 선로나 설비에 흐르는 전력의 분포를 파악하는 것은 현재 계통의 운용이나 미래계통의 설비계획을 검토하는데 매우 중요하다. 정상상태에서 모선의 전압크기와 선로에 흐르는 전력조류의 양을 계산하는 것을 전력조류계산이라고 한다. 조류계산법에 관한 연구는 1950년대부터 수행되어 왔는데[1-3], 초기에는 수치해법 자체에 관심을 두어 빠른 수렴성이나 적은 기억용량으로 대규모 문제를 푸는 방법의 연구가 많았지만[4-8], 최근에는 컴퓨터 사용환경의 변화 및 프로그래밍 언어의 발전에 따라 개발된 소프트웨어의 사용상 편의성이나, 향후 유지보수 문제에 관심을 두고 연구 개발을 하고 있는 추세이다[9-11].

우리나라의 경우, 한국전력공사 등 전력조류계산을 필요로 하는 실무부서에서는 대부분 미국 PTI사의 PSS/E 패키지를 사용하고 있는데, 초기에 Fortran언어로 개발된 PSS/E는 그 기능이 다양하지만 오늘날 컴퓨터의 윈도우 운영체제에는 미흡한 점이 있다.

본 연구에서는 대규모 실계통의 조류계산에 효과적으로 대응할 수 있는 전력조류계산법을 개발하는데 있다. 개발목표로는 첫째, 새로운 윈도우 환경에 적합한 소프트웨어로 개발을 하고, 둘째, 대규모 계통에 적용할 경우 정확한 해를 빠르게 얻을 수 있도록 하며, 셋째, 사용자들이 쉽게 접근할 수 있는 사용자 인터페이스를 가질 뿐만 아니라 기존에 사용되고 있는 소프트웨어와의 입력자료 호환성을 유지하도록 하는데 있다.

이를 위해 본 연구에서는 전력방정식을 풀기 위한 수치해법으로 가우스-자이델법, 뉴턴-랩슨법과 고속분할법을 포함시켰고, 자코비안 행렬의 1차원 배열 기법과 자코비안 행렬의 처리시 나타나는 생성항을 감소하기 위한 모선의 적정배열(Optimal Bus Ordering)기법을 개발하였다. 또한 객체 지향적 프로그래밍(Object-Oriented Programming)기법[9-10]을 도입하여 계통 요소의 추가, 삭제 및 변경시 개체도입을 용이하게 하였다.

개발한 소프트웨어 소프트웨어의 성능을 평가해 보기 위하여 IEEE 14, 30, 57, 118 샘플 계통과 우리나라 실계통을 대상으로 사례검토를 하였으며, 그 결과를 PTI사의 PSS/E와 비교하여 그 효용성을 검토하였다.

2. 전력조류계산

전력시스템의 조류계산에 널리 사용되는 산법으로는 가우스-자이델법, 뉴턴-랩슨법 그리고, 분할법과 고속분할법 등이 있다. 이들 산법 중 가우스-자이델법은 산법이 간단하여 컴퓨터 프로그래밍이 쉽다는 장점이 있지만, 반복 계산량이 많고 수렴속도가 계통의 규모

에 비례하여 증가하는 단점이 있다. 이에 비해 뉴턴-랩슨법은 2차 수렴 특성을 가지기 때문에 가우스-자이델법에 비해 수렴속도가 빠르며, 반복회수가 계통의 규모와 무관(4회~8회)하기 때문에 대규모 계통의 적용에 적합하다. 고속분할법은 뉴턴-랩슨법의 변형된 산법으로서 자코비안 행렬을 상수로 근사화하여 수치해를 구하는 산법이다. 특히 뉴턴-랩슨법 및 그 변형산법들은 모션 어드미턴스 행렬이나 자코비안 행렬 등을 처리할 때 스퍼스 프로그래밍 기법을 적용하고, 자코비안 행렬의 삼각화 과정에서 발생하는 생성항을 감소시키기 위한 행렬 서열화(Bus Ordering)기법을 적용함으로써 계산시간을 더욱 단축시킬 뿐만 아니라 필요한 메모리의 양을 줄일 수 있어 대규모 계통의 적용을 가능하게 한다.

그림 1은 전력시스템의 전형적인 모선을 나타낸 것이며, $y_{11}, y_{22}, \dots, y_{nn}$ 는 선로의 어드미턴스 성분이고, $V_i, V_1, V_2, \dots, V_n$ 는 각 모선의 전압, I_i 는 모션 i 로 유입되는 전류를 나타낸다. 그림 1에서 모션 i 에 대해 *KCL*을 적용하면, 식 (1)과 같이 해당 모션으로 유입되는 전류를 얻을 수 있다.

$$I_i = V_i \sum_{j=0}^n y_{ij} - \sum_{j=1}^n y_{ij} V_j \quad j \neq i \tag{1}$$

한편 모션 i 에서의 유.무효 전력은 식 (2) 또는 (3)과 같이 나타낼 수 있다.

$$P_i + jQ_i = V_i I_i^* \tag{2}$$

또는

$$I_i = \frac{P_i - jQ_i}{V_i^*} \tag{3}$$

이제 식 (1)과 식 (3)으로부터 식 (4)와 같은 전력방정식을 얻을 수 있다.

$$\frac{P_i - jQ_i}{V_i^*} = Y_{ii} V_i - \sum_{j=1}^n Y_{ij} V_j \quad j \neq i \tag{4}$$

여기서 $Y_{ii} = \sum_{j=0}^n y_{ij}$ 이고, $Y_{ij} = -y_{ij}$ 이다.

식 (4)에서 알 수 있듯이 전력방정식은 비선형 대수방정식으로 표현되므로 반복 수치해법에 의해 해를 구해야 한다.

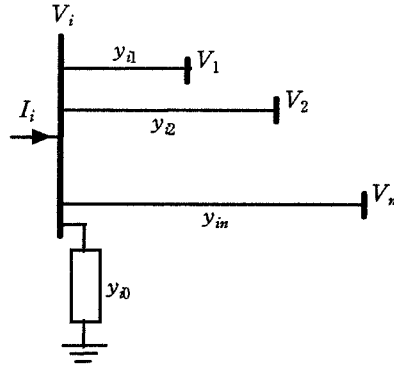


그림 1. 전력시스템의 모선모형

가우스-자이텔법(Gauss-Seidel Method)은 식 (5)와 같은 비선형 방정식을 식 (6)과 같은 형태로 바꾸어 반복계산에 의해 해를 구하는 산법이다.

$$f(x) = 0 \quad (5)$$

$$x = F(x) \quad (6)$$

가우스-자이텔법을 전력조류계산에 적용하기 위하여 식 (4)의 전력방정식에서 전압에 관한 반복식으로 나타내면 다음 식 (7)과 같다.

$$V_i^{(k+1)} = \frac{1}{Y_{ii}} \left[\frac{P_i^s - jQ_i^s}{V_i^{(k)}} - \sum_{j \neq i} Y_{ij} V_j^{(k)} \right] \quad (7)$$

식 (4)의 전력 방정식을 유효 및 무효전력의 향으로 분리하여 표현하면 다음 식 (8)과 (9)를 얻을 수 있다.

$$P_i^{(k+1)} = \text{Re} [V_i^{*(k)} \{ V_i^{(k)} Y_{ii} + \sum_{j=1, j \neq i}^n Y_{ij} V_j^{(k)} \}] \quad j \neq i \quad (8)$$

$$Q_i^{(k+1)} = -\text{Im} [V_i^{*(k)} \{ V_i^{(k)} Y_{ii} + \sum_{j=1, j \neq i}^n Y_{ij} V_j^{(k)} \}] \quad j \neq i \quad (9)$$

뉴턴-랩슨법(Newton-Raphon Method)을 조류계산에 적용시키기 위해 먼저 식 (4)의 전력방정식에서 전압과 모선 어드미턴스를 식(10)과 같이 표현한 후, 유효전력과 무효전력으로 표현하면 식 (11), (12)와 같다.

$$V_i = |V_i| \angle \delta_i = |V_i| (\cos \delta_i + j \sin \delta_i) \quad (10)$$

$$\text{단, } \delta_{ij} = \delta_i - \delta_j, \quad Y_{ij} = G_{ij} + jB_{ij}$$

$$P_i = |V_i| \sum_{j=0}^n ((G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) |V_j|) \quad j=1, \dots, n \quad (11)$$

$$Q_i = |V_i| \sum_{j=0}^n ((G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) |V_j|) \quad j=1, \dots, n \quad (12)$$

만약 전력계통에 n 개의 모선이 있다면, 그 중 1개가 기준모선이고, 부하모선의 수는 n_1 개, 발전기 모선의 수는 n_2 개 라면, $n = n_1 + n_2 + 1$ 이 된다. 식(11)과 (12)를 초기 추정해 근방에서 테일러 급수 전개하여 2차이상의 고차항을 무시하고 행렬로 표현하면 식(13)과 같다.

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \quad (13)$$

식 (13)에서 ΔP 와 $\Delta \delta$ 행렬의 크기는 $(n_1 + n_2) \times 1$ 이며, ΔQ 와 $\Delta |V|$ 행렬의 크기는 $n_1 \times 1$ 이다. 우변 첫 항은 자코비안(Jacobian)행렬이 되며 크기는 $(2n - 2 - n_1) \times (2n - 1 - n_1)$ 이고, 각 행렬 요소들의 크기는 H 는 $(n - 1) \times (n - 1)$, N 은 $(n - 1) \times (n - 1 - n_1)$, M 은 $(n - 1 - n_1) \times (n - 1)$, L 은 $(n - 1 - n_1) \times (n - 1 - n_1)$ 이 된다.

식 (13)에서 계산된 각 모선의 전력 편차 중에서 가장 큰 전력편차가 허용오차 (ϵ)범위 이내에 들면 반복을 종료하고 조류계산은 수렴된 것으로 본다.

$$|\Delta P_i^{(k)}| \leq \epsilon, \quad |\Delta Q_i^{(k)}| \leq \epsilon \quad (14)$$

뉴턴-랩슨법은 산법의 특성상 2차 수렴특성을 갖기 때문에 앞서 소개한 가우스-자이텔 법에 비하여 수렴성이 좋고, 반복회수가 작아 계산시간이 줄어든다는 장점이 있다. 뉴턴-랩슨법에 의한 조류계산의 소요시간은 대부분 매 반복시 마다 수행되는 자코비안 행렬의 구성 및 삼각화 과정에 소비된다.

고속분할법(Fast Decoupled Method)은 계산시간을 감소시키기 위해 계통의 물리적 특성을 고려하여 근사화된 자코비안 행렬을 상수로 지정함으로써 반복초기에 한번만 구성하도록 한 산법이다. 고속 분할법에 의한 조류계산은 산법 자체가 근사화되어 반복회수가 증가하는 단점이 있지만, 각 반복단계에서의 계산시간이 현저히 감소하기 때문에 전체 계산속도는 크게 향상된다. 현재 대부분의 대규모 계통에 대한 조류해석은 고속분할 조류계산법을 많이 사용하고 있으며, 특히 연속조류계산이나 상정사고 해석 등에 유용하게 사용된다.

3. 프로그래밍 기법

3.1 희소행렬의 처리

실제의 전력계통에서 구성되는 자코비안 행렬은 5[%]미만의 비영요소로만 이루어진 희

소행렬이라는 특징을 가지고 있으며, 이러한 희소행렬을 효과적으로 처리하는 프로그래밍 기법이 필요하다. 따라서 스퍼스 프로그래밍의 첫 시도는 비영요소만을 저장함으로써 메모리의 절약 및 계산시간의 절감을 꾀할 수 있다. 또 식 (13)의 계산과정중 계수행렬의 삼각화 시 발생하는 생성항으로 인하여 많은 메모리가 필요하게 되고, 계산시간이 늘어나는 문제가 있다. 식 (13)을 다시 쓰면,

$$Ax = b \quad (15)$$

$$x = A^{-1}b \quad (16)$$

행렬을 삼각화하는 산법으로는 가우스 소거법(Gauss elimination), 가우스-조단 소거법(Gauss-Jordan elimination) 그리고 LU 분해법(LU decomposition) 등이 많이 사용되고 있는데, 본 연구에서는 이들 산법 중 반복계산시 오차가 비교적 적고, 비동차항이 서로 다른 선형 연립방정식을 풀어야 할 경우 가우스-소거법 보다 효과적이라고 알려진 LU 분해법을 적용하였다. LU 분해란 식 (15)에서 계수행렬 A 를 식 (17)과 같이 두 행렬의 곱의 형태로 변형시키는 것이다.

$$A = LU \quad (17)$$

여기서, L 과 U 는 하삼각 행렬과 상삼각 행렬을 각각 의미하며, 식(17)을 이용한 삼각화 과정은 다음과 같다.

$$LUX = b \quad (18a)$$

$$UX = z \quad (18b)$$

$$Lz = b \quad (18c)$$

LU 분해에 의한 삼각화 과정은 그림 2a와 같고, 삼각화로 인한 생성항의 발생은 그림 2b와 같다.

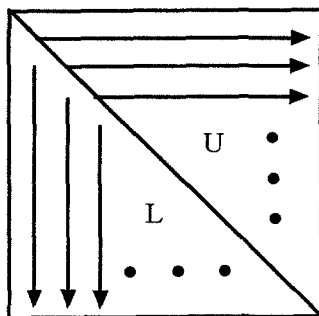


그림 2(a) LU분해과정

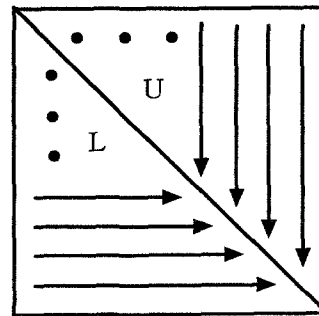


그림 2(b) 생성항 생성과정

그림 2로부터 행렬을 삼각화하는 과정에서 생성항은 행렬의 비대각 요소가 대각선상에서 멀어질수록 생성항이 많아짐을 알 수 있고, R 행 C 열에 위치한 하나의 비대각 요소에 의해 생성되는 생성항의 수 N 은 식 (19)와 같다.

$$N = |R - C| - 1 \tag{19}$$

식 (19)는 상삼각 행렬에서 동일한 열에 비대각 요소가 한 개만 존재할 경우, 혹은 하삼각 행렬에서 동일한 행에 비대각 요소가 한 개만 존재할 경우의 수를 나타낸다. 만약 2개 이상의 비대각 요소가 동일한 열 혹은 행에 위치한다면 생성항의 수는 훨씬 줄어든다.

한편, 계수행렬의 삼각화 시에 발생하는 생성항을 최소화하기 위하여 Tinney 등은 행렬 서열화 기법(Ordering Scheme)을 제안하였다. 하지만 Tinney가 제안한 기법은 소거의 과정에서 연결된 선로의 수가 같은 모선이 발생할 경우 소거될 모선을 임의로 선정함으로써, 비대각 요소가 대각선 근처에 위치하지 않을 수도 있다. 만약 소거의 각 단계에서 연결된 선로의 수가 같은 모선이 발생할 경우 식 (3.4)의 $|R - C|$ 값이 작은 모선을 찾아 우선 소거시키면 생성항을 줄일 수 있다. $|R - C|$ 의 값이 작다는 것은 서로 연결된 두 모선의 모선번호 사이의 차가 작다는 것을 의미하고, 이것은 행렬 구조상으로는 비대각 요소들을 대각선상 근처로 모으는 역할을 한다. 만약 두 모선번호 사이의 차가 1이라면 생성항은 발생하지 않는다. 따라서 본 논문에서는 연결된 선로의 수가 같은 모선이 발생할 경우 최근에 소거된 모선과 연결관계가 있는 모선을 찾아 우선 소거시킴으로써 생성항을 최소화하였다.

이와 같은 내용은 기존의 세 가지 기법 중 Scheme 2와 Scheme 3에 적용할 수 있으며, 이를 요약하면, 아래와 같다.

- 수정된 기법 2: 소거의 각 단계에서, 모선에 연결된 선로의 수가 가장 작은 모선의 순서로 모선 번호를 재배열한다. 이 때 연결된 선로의 수가 같은 모선의 경우, 최근에 소거된 모선과 연결상태가 있는 모선을 우선 소거한다.
- 수정된 기법 3: 각 단계의 소거의 결과, 모선에 연결된 선로의 수가 가장 작은 모선이 먼저 모선번호를 부여받도록 재배열한다. 연결된 선로의 수가 같은 모선의 경우 최근에 소거된 모선과 연결상태가 있는 모선을 우선 소거한다.

기존의 기법들과 마찬가지로 ‘수정된 기법 3’은 계산시간이 ‘수정된 기법 2’에 비하여 오래 걸리는 단점을 안고 있으므로 본 연구에서는 ‘수정된 기법 2’를 적용하여 행렬 서열화를 수행하였다. 그림 3에 기존의 기법 2를 이용하여 IEEE14모선 샘플 계통의 어드미턴스 행렬에 적용한 결과와 기법 2를 수정 적용하여 행렬 서열화 과정을 거친 결과를 나타내었다.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	o	o		o										
2	o	o	o	o	o									
3	o	o	o	x										
4	o	o	o	o				o						
5	o	x	x	o	o	o		x						
6				o	o	x		x	o	o	o			
7				o	x	o	o	o		x	x	x		
8					o	o	x			x	x	x		
9			o	x	x	o	x	o	o	x	x	x	o	
10								o	o	o	x	x	x	
11					o	x	x	x	o	o	x	x	x	
12					o	x	x	x	x	x	o	o	x	
13					o	x	x	x	x	x	o	o	o	
14								o	x	x	x	o	o	

그림 3(a) IEEE14모선의
배열전

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	o						o							
2	o		o		o	x								
3			o	o	o	x	x							
4	o	o	o	o	o	x								
5			o	o	o	x	o							
6	o	x	o	o	o	o	x				o			
7	o	x	x	x	x	o	o					x		
8					o	x	o	o	o		o	x		
9								o	o	o	x	x	o	
10									o	o	x	o	x	
11								o	x	x	o	x	o	
12						o	x	x	x	o	x	o	o	x
13												o	o	o
14									o	x	o	x	o	o

그림 3(b) IEEE14모선의
배열 후(Scheme II 적용-1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	o									o				
2		o								x	o	o		
3			o							x	o	x	o	
4				o	o					x	o	x	x	x
5				o	o				o	x	x	x	x	x
6						o	o	o	x	x	x	x	x	x
7							o	o	x	x	o	x	x	x
8							o	o	o	o	x	x	x	x
9				o	o	x	o	o	x	x	x	o	x	
10	o	x	x	x	x	x	x	x	x	o	o	x	o	x
11				o	x	x	o	x	x	o	o	x	x	o
12	o	o	x	x	x	x	x	x	x	x	o	o	o	
13	o	x	x	x	x	x	x	o	o	x	o	o	o	
14		o	x	x	x	x	x	x	x	o	o	o	o	

그림 3(c) IEEE14모선
배열 후(Scheme II 적용-2)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	o	o													
2	o	o					o	o							
3			o		o		o	x							
4				o	o	o	x	x							
5				o	o	o	o	x							
6				o	o	o	o	o							
7	o	o	x	o	o	o	x						o		
8	o	x	x	x	o	x	o	o	o		o			x	
9								o	o	o	x			x	
10									o	o	x			o	
11								o	x	x	o		o	x	
12												o	o	o	
13													o	o	o
14								o	x	x	o	x	o	o	o

그림 3(d) IEEE14모선
배열 후 (수정 Scheme II 적용)

3.2 객체 지향적 프로그래밍 (OOP; Object-Oriented Programming)

기존의 절차적 프로그래밍 기법은 그 규모가 커질수록 프로그램이 더욱 복잡해짐으로써 개발과정 뿐만 아니라 프로그램의 유지 및 보수가 어렵다는 문제점이 지적되어 왔다. 객체 지향적 프로그래밍 기법이란 기존의 함수 중심의 프로그래밍 기법으로부터 데이터와 그 데이터의 운용(함수 혹은 메서드)으로 이루어진 객체(object)를 중심으로 프로그램을 구현함을 뜻한다. 이러한 객체 지향적 프로그래밍 기법을 전력계통 분야에 적용하기 위한 노력

은 10여 년 전부터 시작되었으며, 현재에도 많은 연구가 진행되고 있다.

전력조류해석에 객체 지향적 프로그래밍 기법을 적용하기 위해서는 먼저 무엇을 객체로 설정할 것인가를 결정하여야 하고, 각 객체의 클래스들 사이의 계층구조를 설계하여야 한다. 전력 조류계산시 전력계통을 구성하고 있는 요소인 모선, 발전기, 부하, 변압기, 선로들을 하나의 객체로 구현할 수 있으며, 또한 조류해석 산법들도 객체로 구현할 수 있다. 본 논문에서 객체 지향적 조류계산을 위해 설계한 클래스 및 그 계층도는 그림 4와 같다.

그림 4에서 'Element'는 계통 내에 포함된 물리적인 구성요소들을 의미하며, 모선, 발전기, 부하, 변압기 등이 포함된다. 그리고 'Element' 클래스는 이들 각 구성요소의 상위 클래스가 된다. 반면, 'LfData' 클래스는 연산을 위한 개념적인 클래스이다. 여기에는 어드미턴스 행렬을 구축하기 위한 'Ybus', 모선 서열화를 위한 'Ordering'등의 메서드(method)를 갖는다.

마지막으로 'Algorithm' 클래스는 하위 클래스로 'Newton-Raphson' 클래스 및 'Gauss-Seidel' 클래스를 갖는다. 특히 'Newton-Raphson' 클래스는 뉴턴-랩슨법에 의한 조류해석을 위한 'Full Newton-Raphson' 클래스와 고속 분할 조류해석을 위한 'Fast-Decoupled' 클래스를 갖는다. 그림 4와 같이 같이 각 클래스들을 계통요소 별로 수평적으로 구성함으로써 부가적인 계통요소가 발생할 경우 기존 코드의 수정 없이 단지 새로운 클래스를 추가할 수 있기 때문에 프로그램의 유연성이 높아진다. 또한 연산을 위한 클래스 "LfData"의 하위 클래스 등도 코드의 재사용성을 확보할 수 있다.

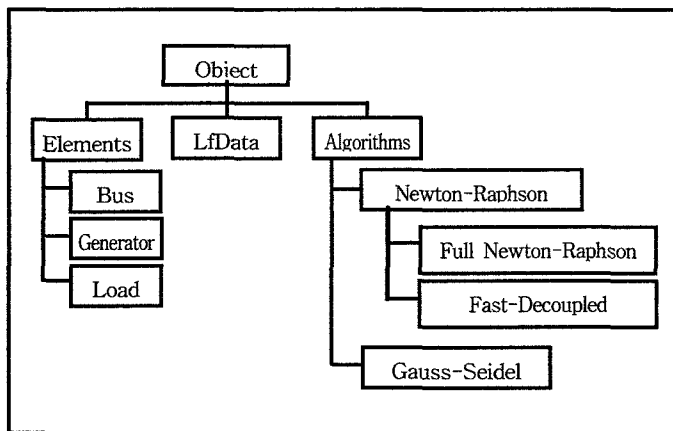


그림 4 조류해석 클래스 및 계층도

4. 사례 검토

본 연구를 통하여 개발한 조류계산 소프트웨어의 성능을 검증해 보기 위하여 IEEE-14, -30, -57 및 -118모선의 샘플계통과 2000년 한국전력의 실계통에 적용하였으며, 그 결과를 PTI사의 PSS/E와 비교하여 보았다. 사례연구에 사용된 컴퓨터의 사양은 MS-Windows 98SE, 펜티움-II MMX450[MHz], 주 메모리 128[MB]이다. 그림 5~7은 각각 IEEE -14, -30, -57 및 -118 모선 샘플계통에 대하여 가우스-자이델법, 뉴턴-랩슨법 및 고속분할법을

적용하여 조류계산을 수행했을 때 각 반복단계에서의 최대 전력편차를 나타낸 것이다. 표 1은 개발된 소프트웨어를 이용하여 각 샘플계통에 대해 3가지 산법을 이용하여 조류계산을 수행한 결과표이다.

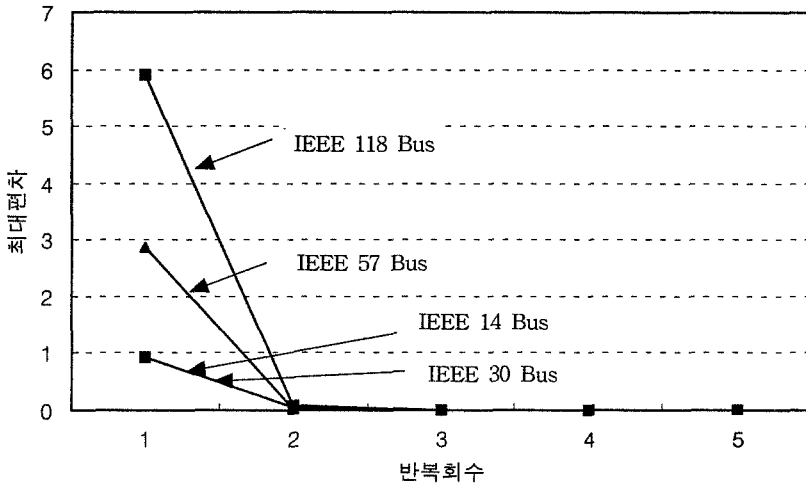


그림 5. IEEE 샘플계통에 대한 뉴턴-랩슨법의 적용

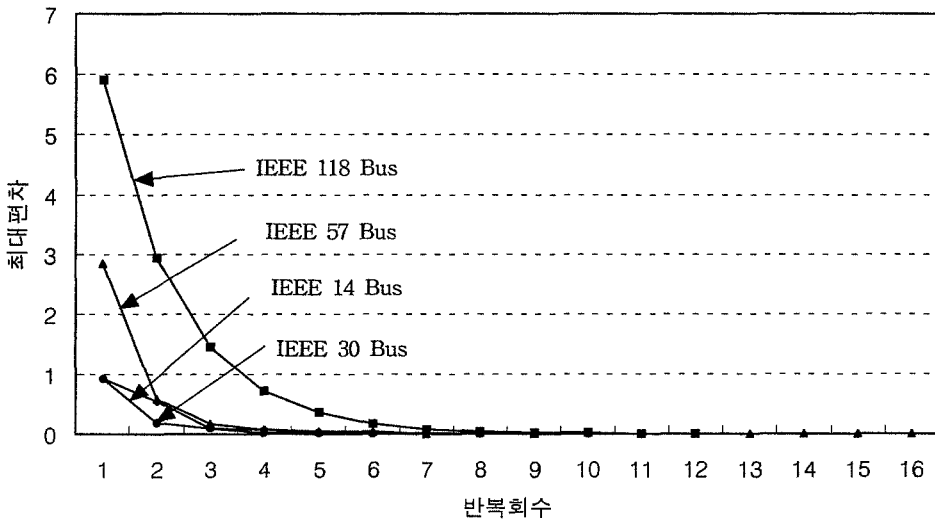


그림 6. IEEE 샘플계통에 대한 고속 분할법의 적용

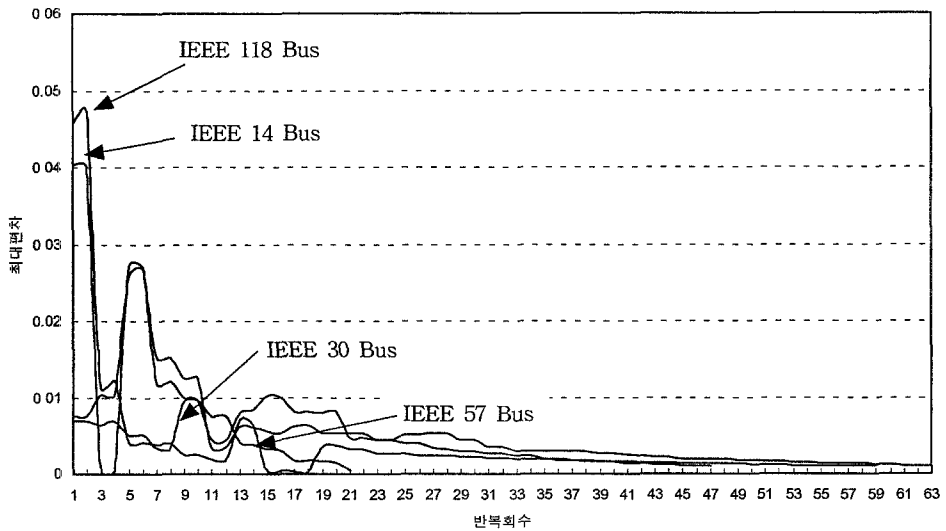


그림 7. IEEE 샘플계통에 대한 가우스-자이델법의 적용

표 1. 각 산법별 조류계산 결과

적용 산법	사 례	반복회수[회]	계산시간[sec]	최대편차
Newton - Raphson	IEEE 14	4	0.00	0.0001
	IEEE 30	4	0.05	0.0001
	IEEE 57	5	0.11	0.0000
	IEEE 118	4	0.17	0.0001
Fast - Decoupled	IEEE 14	10	0.00	0.0048
	IEEE 30	11	0.06	0.0041
	IEEE 57	16	0.07	0.0009
	IEEE 118	12	0.11	0.0042
Gauss - Seidel	IEEE 14	21	0.00	0.0004
	IEEE 30	47	0.06	0.0010
	IEEE 57	63	0.06	0.0010
	IEEE 118	59	0.16	0.0010

표 1에서 계산시간은 데이터를 입력받은 후 수렴에 이르는 시간을 의미하며, 시간의 측정은 계산시간 동안의 CPU 클럭 수를 측정하여 0.01초까지 계산한 값이다.

사례검토 결과, 전력계통이 정상상태인 경우, 뉴턴-랩슨법은 4~8회, 고속 분할법은 10회 안팎의 반복 후 수렴이 되며, 가우스-자이델법은 계통의 규모에 비례하여 반복회수가 증가하므로, 표 1의 결과로부터, 개발된 소프트웨어는 IEEE 시험계통에 대하여 만족할 만한 결과를 얻은 것으로 판단된다.

실계통 데이터는 앞서 언급했듯이 2000년 한국전력 데이터(823 모선)를 사용하였으며,

그 결과를 그림 8 ~10에 나타내었다.

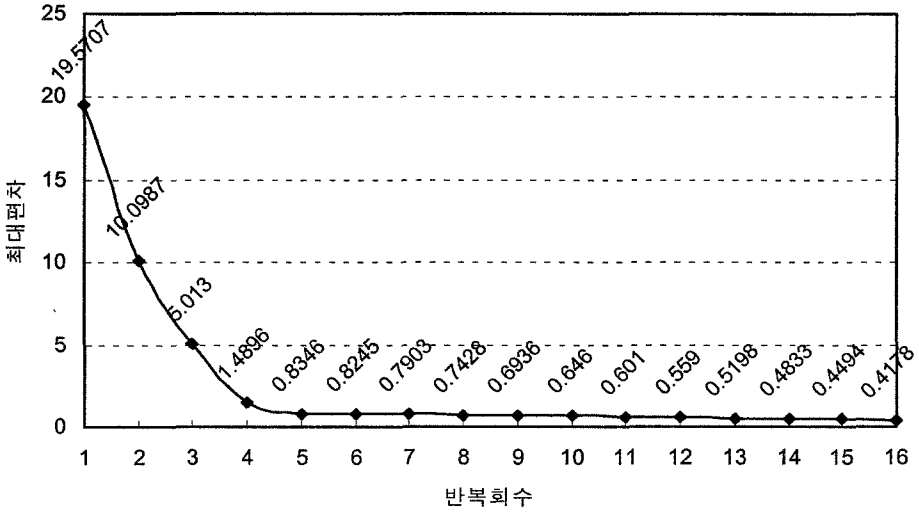


그림 8 한국전력 계통(823모선)에 대한 뉴턴-랩슨법의 적용

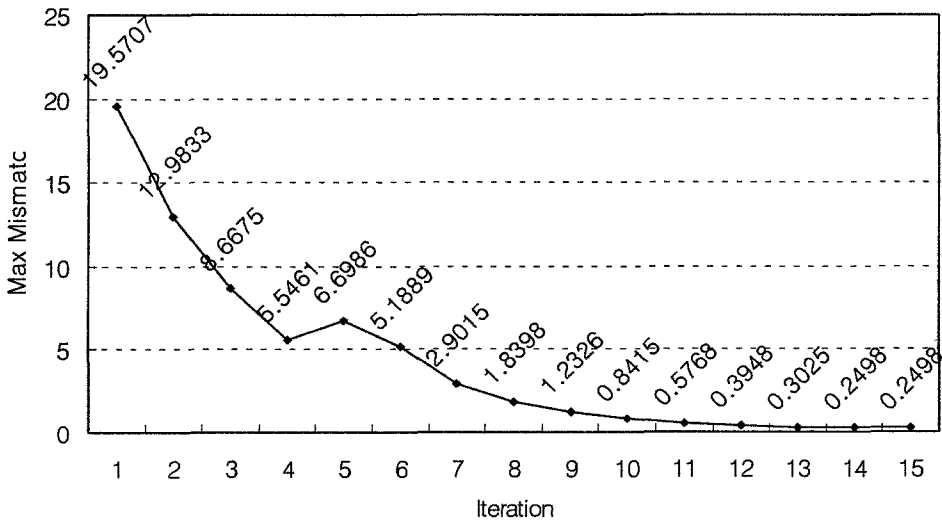


그림 9. 한국전력 계통(823모선)에 대한 고속 분할법의 적용

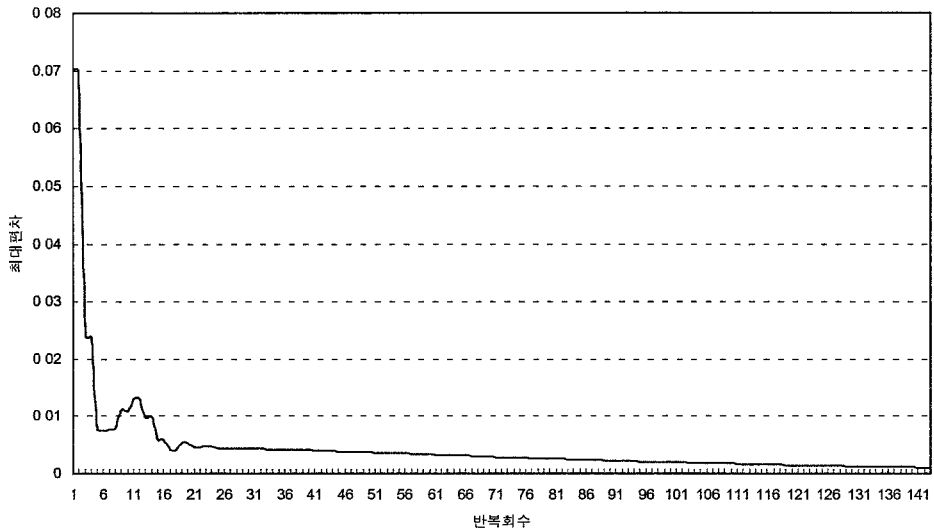


그림 10. 한국전력 계통(823모선)에 대한 가우스-자이델법의 적용

본 연구를 통하여 개발된 소프트웨어와 기존의 전력계통 해석용 상용 소프트웨어인 PSS/E를 이용하여 실제계통 데이터에 대한 조류계산 수행 결과를 표 2에 나타내었다. 표 2에서 'PsaLf'가 본 연구를 통하여 개발된 소프트웨어의 이름이다.

표 5.3 PsaLf와 PSS/E의 조류계산 결과 비교

적 용 산 법	사용 소프트웨어	반복회수[회]	계산시간[sec]	최대편차
Newton - Raphson	PsaLf	16	235.63	0.4178
	PSS/E	16	1.00~2.00	0.0010
Fast - Decoupled	PsaLf	15	9.67	0.2498
	PSS/E	9	1.00~2.00	0.0010
Gauss - Seidel	PsaLf	143	1.54	0.0010
	PSS/E	113	2.00~3.00	0.0010

5. 결론

본 연구를 통하여 새로운 컴퓨터 환경을 고려하면서도 대규모 전력계통의 해석이 가능한 전력조류계산 소프트웨어를 개발하였다. 연구내용을 요약하면 다음과 같다.

- 1) 전력방정식을 풀기 위한 산법으로 가우스-자이델법, 뉴턴-랩슨법 및 고속분할법을 포함하였다.
- 2) 대규모 계통의 문제에 대응하기 위하여 어드미턴스 행렬과 자코비안 행렬의 비영요

소만을 1차원 배열로 저장하였으며, 자코비안 행렬의 삼각화 과정에서 발생하는 생성항을 최소화하기 위해 개선된 모션 서열화 기법을 도입하였다.

- 3) 객체 지향적 기법으로 프로그래밍함으로써 관련 프로그램의 개발시 코드의 재사용을 가능하도록 하였다.
- 4) 기존의 PSS/E 입력양식과 IEEE 입력양식을 모두 사용할 수 있도록 개발하여 프로그램의 유연성을 확보하였다.
- 5) 그래픽 사용자 인터페이스를 통하여 제반 자료의 편집 등의 작업 수행을 용이하도록 하였다.
- 6) 샘플 계통과 실계통의 사례를 대상으로 해의 유효성을 검증하였으며 널리 사용하고 있는 상용화 패키지인 PSS/E의 결과와 비교하였다.

현재 개발된 소프트웨어를 웹 기반에서 실시간으로 해석이 가능하도록 하기 위한 연구와 데이터 입/출력시 그래픽 기능을 추가하여 사용자 위주의 인터페이스를 강화하기 위한 연구를 계속하고 있으며, 추후 개발된 소프트웨어를 확장하여 고장해석이나 안정도해석 기능 등을 포함한 보다 포괄적인 전력시스템 해석용 소프트웨어 패키지로의 확장도 기대된다.

참고 문헌

1. A.F. Glimn and G.W. Stagg, "Automatic Calculation of Load Flow Study", Trans. AIEE(Power Apparatus and System), vol.76, pp817, Oct 1957.
2. B. Stott, O. Alsac, "Fast Decoupled Load Flow", IEEE Trans. on PAS, Vol. 93, pp859-869, May/June, 1974.
3. W. F. Tinney, C. E. Hart, "Power Flow Solution by Newton's Method", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-86, no. 11, pp1449-1967, Nov. 1967.
4. W. F. Tinney, V. Brandwajn, S. M. Chan, "Sparse Vector Method", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-104, No. 2, pp295-301, Feb. 1985.
5. Chun-Liang Lee, Nanming Chen, "Sparse Vector Method Improvements via Minimum Inverse Fill-In Ordering", IEEE Transaction on Power Systems, Vol. 8, no. 1, pp239-245, Feb. 1993.
6. E. C. Ogbuobiri, W. F. Tinney, J. W. Walker, "Sparsity-Directed Decomposition for Gaussian Elimination on Matrices", IEEE Transaction on Power Apparatus, Vol. PAS-89, no.1,, pp141-150, Jan. 1970.
7. W. F. Tinney, "Compensation Method for Network Solution By Optimally Ordered Triangular Factorization", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-91, pp123-127, Jan/Feb. 1972.
8. A. M. Sasson, G. Irisarri, S. F. Hodges, "An Optimal Ordering Algorithm for Sparse Matrix Applications", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-97, no. 6, pp2253-2261, Nov/Dec, 1978.