객체지향기법을 이용한 전력조류계산 및 스파시티 연구

論 文
45~3~1

Load Flow Analysis and Sparsity Study Using Object-Oriented Programming Technique

金 定 年* · 白 榮 植**
(Jung-Nyun Kim · Young-Sik Baek)

Abstract - Power system is becoming more and more complex and large. Existing procedual programming technique can't cope with software flexibility and maintance problems. So, Object-Oriented Programming(OOP) is increasingly used to solve these problems. OOP in power system analysis field has been greatly developed. This paper applies OOP in power flow analysis, and presents new algorithm which uses only a Jacobian to solve mismatch equations, and introduces a new sparse matrix storage method which is different from existing method.

Key Words : Object Oriented Programming, Sparse Matrix, sparsity, Jacobian.

1. 서 론

복잡한 대규모의 전력계통해석 문제를 해석하기 위해서 컴퓨터의 의존도가 더욱더 증가하고 있다. 계통이 복잡하고 대형화함에 따라 해석하는 소프트웨어도 복잡해 가는 추세에 있다. 그러나 전력계통 뿐만 아니라 일반적으로 소프트웨어의 유지보수에 있어서 기존의 프로그램의 확장, 변경및 재사용이 어렵다는 문제점이 대두되고 있다. 뿐만 아니라 기존의 소프트웨어의 재사용의 어려움 때문에 하드웨어발전에 비해 소프트웨어발전은 늦다. 소프트웨어의 유지 및 보수를 하는데 상당한 시간이 소요되며 경비는 전체 소프트웨어 예산의 80%를 차지한다고 한다[1]. 이러한 문제점을 해결하기 위해서 기존의 함수중심의 프로그램에서 객체를 중심으로 프로그램 하는 객체지향 기법이 많이 연구되고 있다.

전력계통 해석의 객체지향기법적용은 현재 그 초기단계에 있다. Foley와 Bose는 전력계통요소 각각을 객체로 모델링하고 그것을 토대로 그래픽 인터페이스의 기본이 되게 하였다. 여기에서 계통요소를 객체화하는데 객체 상호간의 기하학적인 연결상태에 중점을 두고 연구하였다[4, 5]. 그리고 실시간해석이 가능하도록 적용시켰다[7]. Liu와 Shahidehpour도 퍼스널 컴퓨터상에서 객체지향기법을 이용해 그래픽 패키지를 제시하였다[6]. 실제 계통 요소의 객체화가 중점적으로 연구되는 반면 추상적인 행렬을 객체로 표현하려는 노력도 있었다[11]. 본 논문에서는 전력계통의 기본 요소들의 일반적인 모델링을 중심으로 이를 각각의 독립된 객체로 표현하고 다른해석분야의 확장이 가능하도록 하였다. 그 일례로서 계통해석의 가장 기본인 전력조류계산에 적용시켰으며 이 계산에 필수적인 행렬식연산을 위한 스파스행렬에서는 기존의 저장방식과는 다른 저장방식을 택함으로써 객체로 표현하는데 용이하게 하였다. 또한 조류계산의 방식에 있어서는 가우스 사이델법과 고속분할법을 적용시켰다. 고속분할법에 있어서 자코비안 연산에서 하나의 자코비안으로써 연산이 가능하도록 하여 계산속도를 감소시키는 알고리즘을 개발하였다.

*正 會 員 : 慶北大 大學院 電氣工學科 博士課程
**正 會 員 : 慶北大 工大 電氣工學科 敎授 · 工博
接受日字 : 1995年 7月 27日
最終完了 : 1996年 1月 22日

2. 객체지향 프로그래밍 기법

객체지향프로그래밍(Object - Oriented Programming)[12,13]의 주요 목표는 소프트웨어의 확장성과 재사용 가능성을 개선하여 프로그래밍 효율을 향상시키고, 소프트웨어의 유지보수를 용이하게 하며 동시에 개발비용을 줄이고자 하는 것이다.

객체는 어떤 사물의 상태를 나타내는 데이터와 그 데이터의 조작할 수 있는 멤버함수로 나누어질 수 있다. 일반적으로 객체지향 프로그램의 특성은 캡슐화, 계승, 다형성을 들 수 있다.

캡슐화는 데이터와 멤버함수를 묶어 주는 역할로서 특정한 객체만이 자신의 식별자로서 접근하며 데이터 조작이 가능하다. 이는 함수나 데이터를 사용자들이 직접 접근하는 기존의 방식에서 객체에게 메세지를 보내고, 메세지를 받은 객체의 반응을 중심으로 프로그램하며 직접적인 참조에 의한 오류를 방지하며 객체 데이터가 보호될 수 있다.

계승은 선조에게서 물려받는 것으로 이미 만들어진 클래스에서 수정을 가하지 않고 필요한 속성를 계승받고 추가되는 데이터를 정의할 수 있기 때문에 프로그램 소스코드의 길이가 상당히 짧아지며 효율적으로 코드를 관리할 수 있다.

다형성은 서로 다른 객체가 같은 이름의 함수를 사용하는 것으로, 서로 다른 객체가 같은 메세지를 받더라도 서로 다른 반응을 보인다.

3. 계통의 객체 지향적 모델링

전력계통을 구성하는데 기본이 되는 요소로는 모선, 전송선로, 발전기, 부하, 커패시터뱅크등이 있는데 이들의 조합으로 전

체 계통시스템이 구성된다. 이들 구성요소를 객체로 표현하는데 있어서 각 요소들이 가져야 할 특성과 기능을 각각 객체의 데이터와 멤버함수로 표현함으로써 객체로 모델링 할 수 있다.

다음 그림은 각 객체의 심벌을 나타낸 것이다.

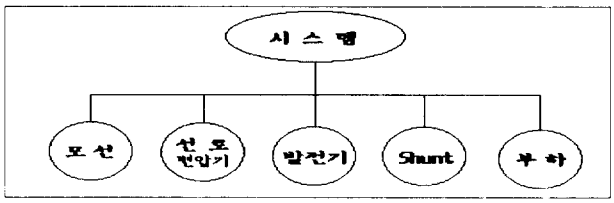

그림 1 전력계통의 구성요소
Fig. 1 Power system components

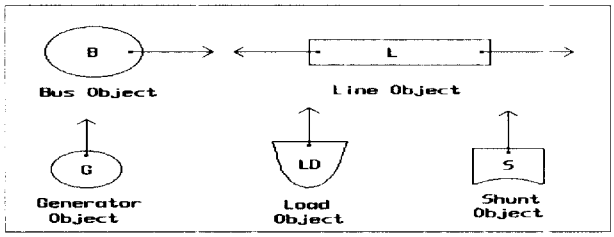

그림 2 개체의 심벌
Fig. 2 Symbols of objects

3.1 각 요소의 객체화

각 객체들 사이의 메세지 전달과 연결 상태를 그림으로 표현하면 다음과 같다. 그림에서 화살표의 방향은 객체들 사이의 메세지 전달 경로 및 연결 상태를 나타내는 것이다.

이상의 객체의 연결상태및 메세지 경로를 통하여 어떠한 한 객체에서 다른 모든 객체로의 메세지 전달이 가능하고 각 객체의 상호 연결시킴으로써 쉽게 계통의 구성을 알 수 있다.

3.2 3모선 3선로 계통의 객체화

그림 7은 그림 6의 간단한 3모선 3선로 시스템을 객체의 심벌로서 나타낸 것이다.

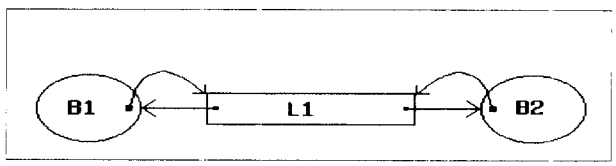

그림 3 모선-선로의 메세지 경로
Fig. 3 Message flow between bus and line

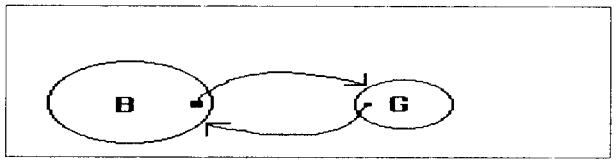

그림 4 모선-발전기의 메세지 경로
Fig. 4 Message flow between bus generator

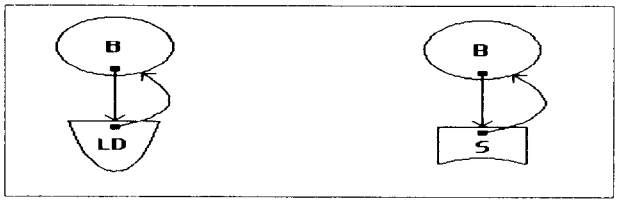

그림 5 모선-부하 및 모선-Shunt의 메세지 경로
Fig. 5 Message flow between bus and load, shunt

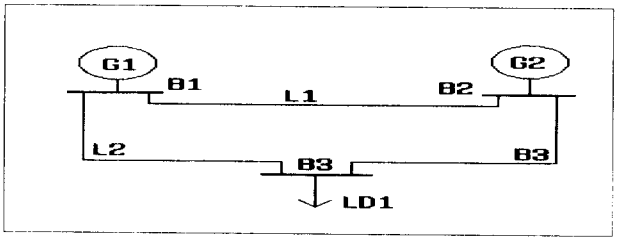

그림 6 3모선 3선로 시스템의 단선도
Fig. 6 Single line diagran of 3 bus and 3 line

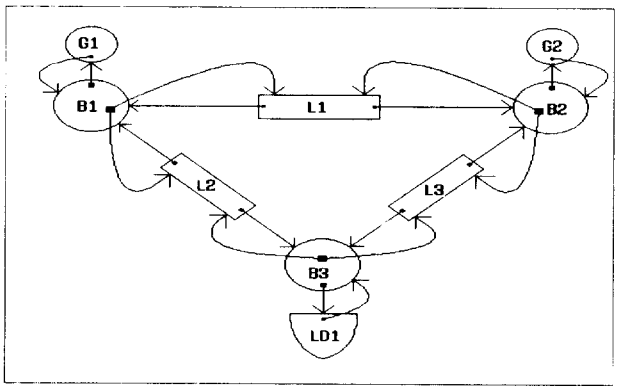

그림 7 시스템의 객체화
Fig. 7 Object representation of a system

4. 객체지향 모델의 이용예

4.1 사고시의 모델링

그림 8은 그림 7에서 선로 3에서 지락사고가 발생했을 때의 과정을 나타낸 것이다.

선로 3에서 사고시 사고가 발생한 지점을 사고모선으로 간주하고 새로운 모선 FB로 새로이 생성된다. 그리고 사고지점을 중심으로 임피던스가 분리되며 선로 LF1과 LF2가 생성되고 사고모선과 연결된다. 그리고 사고가 발생한 선로 3은 소멸되며 새로운 계통이 형성된다.

그림 9에서는 사고 선로가 소멸하는 모습을 선로와 모선을 중심으로 표현한 것이다.

소멸될 선로 객체 LF는 선로가 연결된 두 모선객체 B2, B3에게 메세지를 보내 메세지의 경로가 되는 1과 2를 제거하고, 자신이 모선에게 메세지를 보내는 경로 3, 4를 제거한다.

이렇게 함으로써 모선 객체 B2, B3은 LF에 메세지 전달을 하지 못하도록 한다. LF또한 B2, B3에게 메세지 전달을 하지 못한다. 마지막으로 선로객체 LF은 소멸한다.

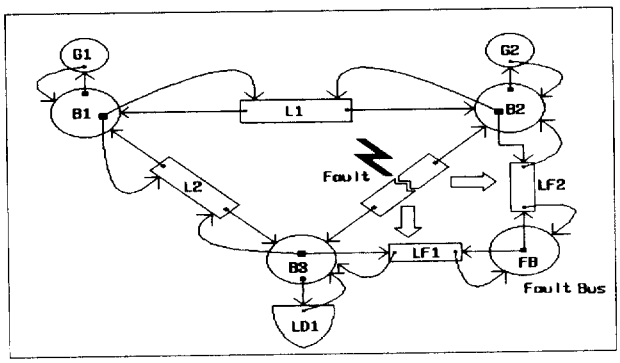

그림 8 사고시 객체의 변형(선로 3에서 사고)
Fig. 8 Object modification in fault case at bus 3

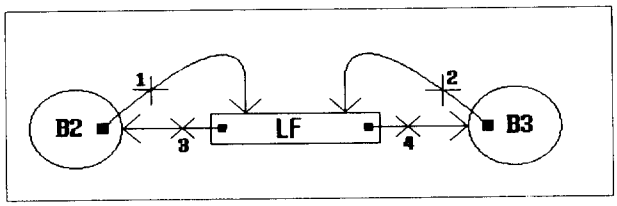

그림 9 선로의 소멸
Fig. 9 Destruction of the line

이렇게 객체들이 독립적으로 존재함으로써 다른 객체들에게는 영향을 미치고 않고 계통 시스템을 유지할 수 있다. 선로가 생성될 경우에는 반대의 순서를 따르면 된다.

기존의 방식 절차위주(Procedual) 프로그램의 경우 선로가 제거되는 경우 전 시스템을 새로이 구성해야 하는데 비해 이 경우는 한 객체만을 제거함으로써 전 시스템에 영향을 미치지 않도록 구성할 수 있다. 이 경우를 IEEE14모선 시스템을 시뮬레이션하고 그 시간을 비교하여 보았다.

선로에서 사고가 발생하고 그 때문에 계통을 새롭게 구성해서 조류계산을 행하는데 걸리는 시간을 비교해 보았다. 절차위주 프로그램의 경우 계산시간(T_p)은 0.299초인데 비해 객체지향 기법(T_o)에서는 0.217초의 결과가 나왔다.

기존의 방법과 제안한 방법의 비 T_o/T_p=0.726이다. 이는 계통이 변했을 경우 새로운 시스템을 구성하기 위하여 드는 시간의 차이가 나기 때문이다. IEEE30모선의 경우 T_o/T_p=0.85로 시간을 단축할 수 있다.

4.2 가우스 사이델법을 이용한 전력조류계산

가우스 사이델법을 이용한 전력조류계산에서 어떤 모선에서의 전압수정식은 다음과 같다.

$$V_i^{(k+1)}=\frac{1}{Y_{ii}}\left(\frac{S_i^*}{V_i^{(k)*}}-\sum_{j=1}^{n}Y_{ij}V_j^{(k)}\right)\ [\,i\neq j] \tag{1}$$

$$S_i=P_i+jQ_i \tag{2}$$

$$Q_i=-Im[\,(Y_{ii}V_i+\sum_{j=1}^{N}Y_{ij}V_j)V_i^*] \quad [\,i\neq j] \tag{3}$$

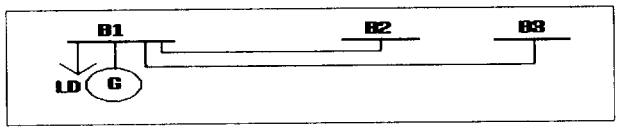

그림 10 모선에서의 전압수정
Fig. 10 Voltage update in a bus

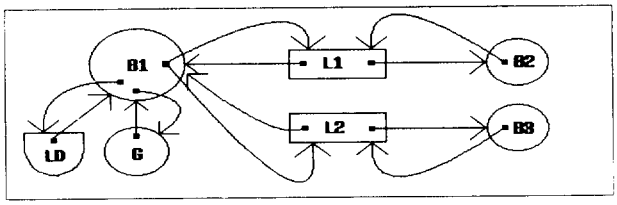

그림 11 전압수정시 메세지 경로
Fig. 11 Message flow in case of a boltage update

"전압수정하라"는 메세지를 받은 B1모선객체는 다음과 같이 반응한다.

S_i를 구하기 위해서는 모선에 연결된 발전기 G객체, 부하 LD객체에 메세지를 보내서 모선 B1으로의 유입전력을 구한다. 만약 모선이 PV모선인 경우 Q_i를 계산하기 위해 선로객체L1, L2에 메세지를 보내 계산한다.

(식.1)에서 $\sum_{j=1}^{n}Y_{ij}V_j$항을 구하기 위해 모선에 연결된 모든 선로객체 L1, L2에 메세지를 보내 선로에서 유입전류를 구하라는 메세지를 보낸다. 메세지를 받은 선로객체L1, L2는 전류를 계산하기 위해 메세지가 온 모선B1 뿐만 아니라 반대편 모선(B2,B3)의 전압을 알아야 하기 때문에 두 모선(B2,B3)으로 메세지를 보내고 그 값을 받은 후 선로의 전류를 계산하고, 계산된 값을 모선객체에 준다. 모선에 연결된 여러 객체들에게 메세지를 보내 전압 수정식에 의해 계산한 후 마지막으로 다음 모선객체에 "전압을 수정하라"는 메세지를 보냄으로써 해당모선의 전압은 수정된다. 표 1은 이 방법에 의한 샘플시스템의 계산 시간을 나타낸다.

가우스 사이델법은 객체지향 프로그램으로 구성할 경우 이해하기가 쉽고 객체의 개념을 적용시키기는 쉽지만 모선수가 많으면 반복수가 급격히 증가할 뿐만 아니라 수렴시간 또한 증가하기 때문에 단일 프로세서에서는 효과적이지 못하다.

이러한 가우스 사이델법의 수렴시간 문제를 해결하기 위해서는 다중 프로세서로써 극복할 수 있다. 객체들 상호간에 독립성이 유지되고 한 프로세서에 한 객체를 할당하고 각 프로세서가 병렬처리된다면 수렴시간은 대규모 계통에서도 사용가능할 정도가 될 것이며 이 방법에 대한 계속적인 연구가 필요하다.

표 1 가우스 사이델법의 수렴시간
Table 1 Convergence time of gauss-seidel method

시스템	IEEE 14	IEEE 30
수렴시간[sec]	0.1875	1.465

5. 객체지향 기법을 이용한 뉴튼랍슨 전력조류계산

5.1 스파스 행렬의 객체화

일반적으로 전력 계통에서 하나의 모선은 인접 2~3개 모선

과 연결을 가지므로 Y_{Bus}는 '0'이 아닌 요소보다 '0'인 요소가 훨씬 많다. 이 경우 행렬은 소(sparse)하다고 하고, 그 행렬을 소행렬이라고 한다. 실제 전력계통의 경우 '0'이 아닌 요소는 전체 요소의 2~3%에 불과하다.

$A=$

d1	a	b	c	
	d2		d	
	e	d3		f
		g	d4	
	h	i		d5

기존의 스파스행렬의 일차원 배열을 사용하는 기억방식은 객체로 구성할 때 복잡해지고, 대각요소및 비대각요소를 따로 기억시켜므로 행렬전체가 하나의 객체라는 의미가 없다. 또한 값의 삽입 및 삭제의 경우 번거로움이 발생한다. 그런 단점을 보완하기 위해 새로운 기억방식을 택함으로서 그 문제점을 해결하였다.

위의 A행렬을 저장하는 모델을 만들면 다음과 같다.

그림 12의 저장모델을 이용해 객체를 구성한다면 행렬요소의 값의 변경, 삽입, 삭제가 간단하며 행렬의 크기 또한 변경이 용이하다.

연산자 중복정의에 의한 ( )연산자로써 행렬요소값의 할당, 대입 및 변경, 삽입을 동시에 할 수 있다.

표 2는 괄호( )연산자의 사용 예이다.

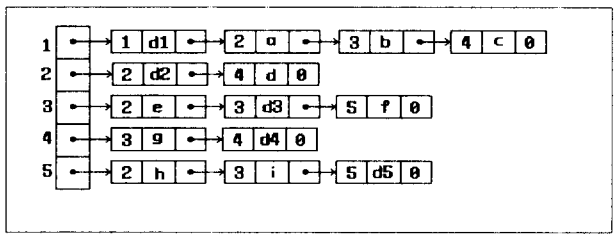

그림 12 스파스 행렬의 저장

Fig. 12 Storage scheme of sparse matrix elements

표 2 연산자 괄호( ) (그림12 참조)

Table 2 ( ) operator usage

사용예	역 할
A(5,4) = 0.03	5행 4에 0.03을 할당
A(5,4) = 15.4	5행 4열의 값이 0.03이 15.4로 변경
x = A(5,4)	5행 4열의 값(15.4)이 x에 할당
x = A(2,5)	2행 5열의 0이 x에 할당

5.2 자코비안 연산

고속 분할법의 경우 다음과 같은 Jacobian연산이 필요하다.

$$\left[\frac{\triangle P}{|V|}\right]=[B_P][\triangle\theta] \tag{4}$$

$$\left[\frac{\triangle Q}{|V|}\right]=[B_Q][\triangle|V|] \tag{5}$$

B_P : $Y_{Bus}=G_{Bus}+jB_{Bus}$의 $[-B]$ 행렬
(NPV+NPQ)×(NPV+NPQ)

B_Q : $Y_{Bus}=G_{Bus}+jB_{Bus}$의 $[-B]$ 행렬
(NPQ)×(NPQ)

NPV : PV모선수

NPQ : PQ모선수

위의 B_P자코비안요소는 조류계산과정에서 변하지 않으므로 처음에 한번만 구해 놓으면 다시 구할 필요가 없다. 그러나 B_Q자코비안의 경우 무효전력의 상·하한치가 벗어난 B_Q의 자코비안의 역행렬을 다시 구해야 한다.

제안한 방법에서는 B_Q^{-1}을 구하지 않고 변하지 않은 B_P^{-1}를 이용해 무효전력의 편차방정식을 구하였다. 다음 두 방법에서 PV모선의 ΔV를 0으로하는 ΔQ를 구한다.

<방법 1>

$$[\triangle|V|]=[B_P]^{-1}\left[\frac{\triangle Q}{|V|}\right] \tag{6}$$

$$[A]=[B_P]^{-1} \tag{7}$$

행렬A (NPV+NPQ)×(NPV+NPQ)

PV모선인 한행을 살펴보면 다음과 같다.

$$\triangle|V_i|=\sum_{j=1}^{NPQ}A_{ij}\frac{\triangle Q_j}{|V_j|}+\sum_{k=1}^{NPV}A_{ik}\frac{\triangle Q_k}{|V_k|}=0 \tag{8}$$

$$\sum_{j=1}^{NPQ}A_{ij}\frac{\triangle Q_j}{|V_j|}=-\sum_{k=1}^{NPV}A_{ik}\frac{\triangle Q_k}{|V_k|} \tag{9}$$

위 식에서 좌변은 반복과정에서 알 수 있는 기지항들이고 우변은 미지항들이다.

$T_i=\sum_{j=1}^{NPQ}A_{ij}\frac{\triangle Q_j}{|V_j|}$ 라 하면

$$T_i=-\sum_{k=1}^{NPV}A_{ik}\frac{\triangle Q_k}{|V_k|} \tag{10}$$

$$T=[A_0]\left[\frac{\triangle Q}{|V|}\right] \tag{11}$$

행렬A_0 (NPV)×(NPV)

에서 PV모선에서의 무효전력의 변동분을 계산해 낼 수 있다.

행렬 B_Q[NPQ×NPQ]대신 A_0[NPV×NPV]를 풀어야 한다.

일반적으로 조류계산에서 PV모선이 차지하는 비율이 대략 20%정도이다[8]. 이 방법은 PV모선수가 적은 계통에 적합 할 것이다.

<방법 2>

다음은 발전기 모선을 포함한 무효전력의 편차방정식이다.

$$\left[\frac{\triangle Q}{|V|}\right]=[B_P][\triangle|V|] \tag{12}$$

위 방정식에서 한 행(row)만을 살펴보면 다음과 같다.

$$\frac{\triangle Q_i}{|V_i|}=\sum_{j=1}^{NPQ}B_{ij}\triangle|V_j|+\sum_{k=1}^{NPV}B_{ik}\triangle|V_k| \tag{13}$$

위 식에서 PV모선의 무효전력의 변화분[$\triangle Q$]을 계산해낸다. $\triangle Q_0$, $\triangle Q_1$의 전력편차가 생겼을 경우 모선의 전압의 크기의 변화를 $\triangle V_0$, $\triangle V_1$라 할 때 다음과 같은 선형관계가 성립한다고 가정하면 다음과 같다.

$$\triangle V=[\frac{\triangle V_1-\triangle V_0}{\triangle Q_1-\triangle Q_0}](\triangle Q-\triangle Q_0)+\triangle V_0 \tag{14}$$

$\triangle Q_0=0$, $\triangle Q_1=1$라 하면

$$\triangle V=[\triangle V_1-\triangle V_0]\triangle Q+\triangle V_0 \tag{15}$$

여기서 PV모선에서의 전압의 변화($\triangle V$)는 0 이므로

$$\triangle Q = -\frac{\triangle V_0}{\triangle V_1-\triangle V_0} \tag{16}$$

가된다.

5.3 계산결과의 비교

표 3은 기존의 고속분할뉴튼랍슨법 과 제안한 두 가지 방법을 비교한 것이다.

표 3 수렴시간 비교

Table 3 Convergence time of fast decopled method

방법 \ 모선		IEEE14 모선		IEEE30 모선	
		수렴시간	sec/iter	수렴시간	sec/iter
기존의 방법		0.1667	0.01515	0.9833	0.08939
제안한 방법	1	0.0833	0.00926	0.1667	0.01515
	2	0.0333	0.00370	0.1333	0.01212

위와 같은 결과는 기존의 방법은 발전기 무효전력이 상하한치를 벗어난 경우 자코비안의 구성과 역행렬을 구하는데 시간이 걸리기 때문인 것으로 생각된다.

6. 결 론

이상과 같이 객체 지향적인 프로그램밍기법을 사용함으로써 기존의 방법에 비교해 볼 때 다음과 같은 점이 개선되었다.

(1) 각각의 전력계통 요소를 독립된 객체로서 표현함으로써 계통 구성요소의 파악이 용이하게 하였으며 다른 전력계통 해석분야의 이용가능성이 클 것이다. (또한 본 연구에서 사용된 객체표현 방법을 아무수정없이 고장계산과 안정도 해석 등에 적용시켜 본 결과 이용이 쉬움이 입증되었다.)

(2) 객체지향 프로그래밍 기법을 전력조류계산에 적용시키고 가우스 사이델법 및 뉴튼랍슨법을 적용시켰다.

(3) 새로운 행렬의 저장방식으로 행렬객체를 구성하였으며 다른 해석에도 이용 가능성이 클 것이다.

(4) 고속 분할법에 있어서 자코비안을 하나를 사용함으로써 Memory면에서 낭비를 줄이고 수렴속도가 개선되었다.

본 논문은 한국전력공사의 연구비 지원에 의해 기초전력 공학공동연구소 주관으로 수행된 과제임(과제번호 94-005)

참 고 문 헌

[1] Guido Buzzi-Ferraris, "Scientific C++ Building Numerical Libraries the Object - Oriented Way", Addison-Wesley Publishing Company, 1993. pp, 364~381.

[2] 이재용, "컴퓨터그래픽기능 이용한 전력조류패키지의 개발", 경북대학교 대학원, 1992.

[3] J. Arrillaga, C. P. Arnold, B. J. Harker, "Computer Modelling of Electriccal Power Systems", John Wiley & Sons Ltd, 1983. pp, 4~42.

[4] M. Foley, A. Bose, W. Mitchell and A. Faustini, "An Object Based Graphical user Interface for Power Systems", IEEE Transactions on Power Systems, Vol. 8, No. 1, Feb. 1993, pp.97-104

[5] M. Prais and A. Bose, "A Toplology Processor that Tracks Network Modification Over Time," IEEE Transactions on Power Systems, Vol. 3, No. 3 Aug 1988, pp 992-998

[6] S. Liu, S.M. Shahidehpour, "An Object-Oriented Power System Graphics Package for Personal Computer Environment," IEEE Summer Power meeeting, Seattle, WA, July, 1992.

[7] M. Foley, A. Bose, "Object-Oriented On-Line Network Analysis", IEEE Transactions on Power Systems, Vol. 10, No. 1 Feb 1995, pp 125-131

[8] Charles A. Gross, "Power System Analysis" John Wiley & Sons Ltd, 1986. pp,255~363.

[9] Curtis F. Gerald, Patrik O. Wheatly, "Appiled Numeracal Analysis", Addison-Wesley Publishing Company, 1989. pp, 153~156,158~163.

[10] A. F. Neyer, F. F. Wu, K. Imhof, "Object-Oriented Programming For Flexible Software:Example of A Loadflow," IEEE Transactions on Power Systems, Vol. 5, No. 3 Aug 1990, pp 689-696

[11] B. Hakavik, A. T. Holen, "Power Sysyem Modeling And Sparse Matrix Operations Using Object-Oriented Programming," IEEE Transactions on Power Systems, Vol. 9, No. 2 May 1994, pp 1045-1051

저 자 소 개

김 정 년 (金 定 年)

1969년 2월 21일생. 1994년 2월 경북대 전기공학과 졸업. 1996년 2월 경북대 대학원 전기공학과 졸업(석사). 현재 경북대 대학원 전기공학과 박사과정

백 영 식 (白 榮 植)

1950년 7월 8일생. 1974년 서울대 전기공학과 졸업. 1977년 서울대 대학원 전기공학과 졸업(석사). 1984년 서울대 대학원 전기공학과(공박). 1977년~1984년 명지대 전기공학과 조교수. 현재 경북대 전기공학과 교수