2000년도 대한전기학회 하계학술대회 논문집 2000.7.17-20

Newton-Raphson법을 이용한 조류계산을 위한 효율적인 LU분해 계산 방법에 관한 연구

김재현, 이소영
국립 순천 대학교

A Study of the effective method of LU factorization for Newton-Raphson Load Flow

Jaehyeon Gim, Soyoung Lee
Sunchon National University

Abstract - This paper introduces new ordering algorithms using the graph of data structure and forward/backward substitution of LU decomposition using recursive function. The performance of the algorithm is compared with Tinney's algorithm using 14 bus systems. Test results show that the new fill-in element of Jacobian matrix using the proposed ordering algorithm is same as that of Tinney scheme 3 and the forward/backward substitution can reduce the computation time

1. 서 론

전력계통의 계획 및 운용에 필수적으로 조류계산은 필요하다. 조류계산은 비 선형 전력방정식을 구성하고 해로서 전압의 크기와 위상을 구하는 것으로 수치해석적인 방법인 Gauss-Seidel법과 Newton-Raphson법이 주로 사용된다. 비 선형 전력 방정식을 선형화하여 초기값으로 반복 계산하여 해를 구하는 방법이다. 특히 Newton-Raphson법은 초기값이 해로 이동하는 변화분은 Jacobian행렬의 역행렬을 구하여 계산되는데 이 계산에 사용되는 기법이 LU로 분해하고 전진 대입법과 후진대입법을 통하여 근을 구한다. 이 Jacobian행렬은 행렬 요소가 희소하며, 대칭적인 위치에 요소의 값을 가지는 특성을 가지고 있다. 이 행렬의 LU분해와 전,후진 대입법의 계산 속도를 증진하기 위하여 모선들의 최적 순서 배치 기법들이 연구되었다. Tinney와 Walker에 의하여 개발된 MD(Minimum Degree)와 MF(Minimum Fill-in)를 이용한 LU분해를 위한 최적 순서(Optimal Ordering) 기법이 소개되었다.[1] Betancourt는 부분 행렬을 최소 경로 길이를 이용한 재분해를 위한 순서를 제안하였다[2]
본 논문에는 조류계산에 사용되는 Jacobian 행렬을 데이터 구조의 하나인 동적 연결 리스트(Linked-List)로 구성하고 그래프(Graph) 데이터 구조를 이용하여 최적 순서를 결정하는 기법과 LU로 분해후 전·후진 대입법은 순환 호출(recursive)기법을 이용하여 효과적인 계산 방법을 제시한다.

2. 최적 순서(Ordering) 기법

2.1 그래프(Graph) 이론

그래프는 마디(Node or Vertices)와 마디를 연결하는 가지(Line or Edge)로 구성된다. 완전 그래프(Complete Graph)는 n개의 마디가 있을 때 $n(n-1)/2$개의 가지(edge)를 가지고 있으면 이를 완전 그래프라 한다. 다시 말하면 한 모선에 연결되어 있는 다른 모선들이 서로 모두 연결되어 있다면 그 모선들만으로 이루는 그래프는 완전 그래프이다. 예를 들면 그림 1의 모선 ① ② ③에서 모선②는 완전 그래프를 구성하고 있으며 모선 ① ③은 완전 그래프가 아니다.

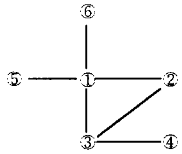

그림 1 그래프

전력계통의 모선이 완전 그래프의 조건을 만족하기 위한 가지의 수가 적은 순으로 모선을 선택하고 선택된 모선은 계통에서 제하고 다시 완전 그래프의 조건을 확인하여 오더링을 하면 LU분해시 완전 그래프를 구성하기 위한 가지 수만큼의 행렬의 새로운 요소가 생기게 된다. 이와 같이 각 모선을 선택하고 반복적으로 모선 순서를 정한다.

3. 순환 호출(Recursive)

함수중에 자기 자신을 호출하는 것을 순환 호출이라 하며 자기 자신으로부터 호출이 되는 함수를 순환 함수(Recursive Function)라 한다.
일반적으로 순환 호출을 이용하면 시스템의 위험성을 희생하고 실행 시간이 단축 할 수 있고 프로그래밍을 간결하게 하는 장점이 있다. 특히 Jacobian 행렬을 LU로 분해된 행렬은 상 삼각 행렬과 하 삼각 행렬이 하나의 행렬로 구성되어 있어 효과적으로 순환 호출기법을 이용할 수 있다.

3.1 Jacobian 행렬의 표현

Jacobian 행렬은 희소 행렬이므로 연결 리스트를 이용하여 그림 2와 같이 구성하였으며 행렬 계산에 필요한 함수들은 글래스의 메소드 함수로 구성하였다.[3]

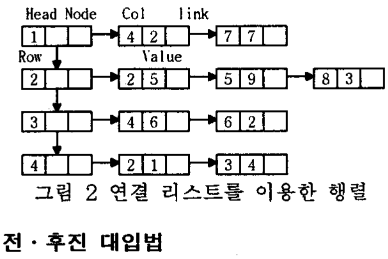

그림 2 연결 리스트를 이용한 행렬

3.2 전·후진 대입법

그림 2와 같은 Jacobian 행렬은 모든 대각 요소는 하삼각 행렬에 포함하며 상 삼각 행렬의 대각 요소는 1이 되는 LU행렬로 분해하였다. Jacobian행렬식을 식(1)과 같다.

$$Jx = [L \backslash U]x = b \tag{1}$$

식(1)을 식(2)와 식(3)으로 분리하여 근을 구한다.

$$Ly = b \tag{2}$$

$$Ux = y \tag{3}$$

식(2)의 y을 구하는 것이 전진 대입법이고 식(3)의 x를 구하는 것이 후진 대입법이다. 우선 전진 대입법으로 모든 y의 값을 구한 후 후진대입법으로 x를 구하게 된다. 그러므로 전진 대입법으로 행렬 J의 i행으로 y_i을 구하고 다음 행을 순환 호출한다. 마지막 행의 전진 대입법을 수행하고 후진 대입법은 순환 호출 함수에서 되돌아온 값을 이용하여 수행한다. 그러므로 일반적인 프로그램에서는 전진 대입법을 수행한 후 후진 대입법을 수행할 때 행렬의 시작 요소, 즉 LU행렬의 대각 요소를 다시 찾아서 실행을 해야 하는데 반하여 순환 호출 기법을 이용함으로서 이러한 작업을 줄일 수 있어 계산시간을 단축할 수 있다.

4. 모의 실험

본 논문에서는 14모선에 15선로 계통인 모의 계통을 실험하였다. 이 계통에서 ⑦ ⑨ ⑩ ⑫ ⑭모선은 완전 그래프 조건을 만족하고 있다.

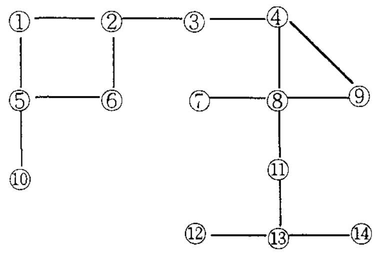

그림 3 모의 실험 계통

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

그림4 Tinney의 기법 1

참고문헌[1]에서 개발한 오더링 기법을 그림3의 모의 실험 계통에 적용한 결과이다. 그림4, 그림5, 및 그림6은 Tinney의 기법 1,2,3이다. 여기서 x은 기존에 있은 요소이고 o은 LU분해시 새로 생성되는 영(0)이 아닌 요소로서 기법 1에서는 10개가 새로이 생성되었고, 기법2에서는 4개, 기법3에서는 2개가 생성되었다. 이 결과에 따르면 기법 3이 가장 좋은 방법이나 프로그램 로직이 복잡하다. 본 논문에서 제시한 그래프를 이용한 기법은 Tinney 기법3과 같은 결과를 얻었다.

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

그림 5 Tinney의 기법 2

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

그림 6 Tinney의 기법3

순환 호출을 이용한 전후진 대입법은 IEEE의 30모선 계통을 모선번호순으로 오더링된 Jacobian행렬을 대상으로 100회 반복적으로 계산된 시간을 비교하였다. 순환 호출을 이용한 방법은 0.297초이었으며 그렇지 않은 방법은 0.343초로서 순환호출을 이용한 방법이 짧았다.

3. 결　　론

본 논문에서는 데이터 구조의 그래프 원리를 이용한 오더링 기법과 순환 호출을 이용한 전후진 대입법을 조류계산의 Jacobian행렬 계산에 적용하였다. 그래프를 이용한 오더링 기법은 행렬의 새로운 요소가 Tinney의 기법3과 같은 수가 생성되었으며 순환 호출을 이용한 전후진 대입법은 전통적인 방법보다 계산 시간을 단축할 수 있었다.

[참 고 문 헌]

[1] William. F. Tinney, John W. Walker, "Direct Solution of Sparse Network Equations by Optimally Ordered Triangular Factorization" Proceedings of the IEEE vol. 55, no. 11 pp1801-1809 Nov. 1967

[2] Ramon Betancourt, "An Efficient Heuristic Ordering algorithm for Partial Matrix Refactorization" IEEE trans. on PS, vol.3, no.3, pp1181-1187 Aug. 1988

[3] Sartaj Sahni, "Data Structures, Algorithms, and Applications in C++," McGraw-Hill 1998