

자기설계 - 기말 보고서

: Advanced Power Flow Capstone Project

Jacobi 팀

융합전자공학전공 201910906 이학민

[Contents]

I . Introduction	2
1) 프로젝트 개요	2
2) 프로젝트 진행 일정	3
3) 자기설계학점 행정 사항	3
II . Body	4
1) Matrix Calculation 모듈 개발	4
2) 선형방정식의 Iterative Solution 개발 및 성능 비교	7
3) 전력 조류 계산 알고리즘 개발	11
4) 다양한 전력 계통의 raw data 제작	19
5) 알고리즘 성능 검증 및 결과 비교	26
III . Conclusion	41
1) 프로젝트의 한계 및 극복방안	41
2) 프로젝트 수행 소감 및 기대효과	41
3) 향후 계획	41

I. Introduction

1) 프로젝트 개요

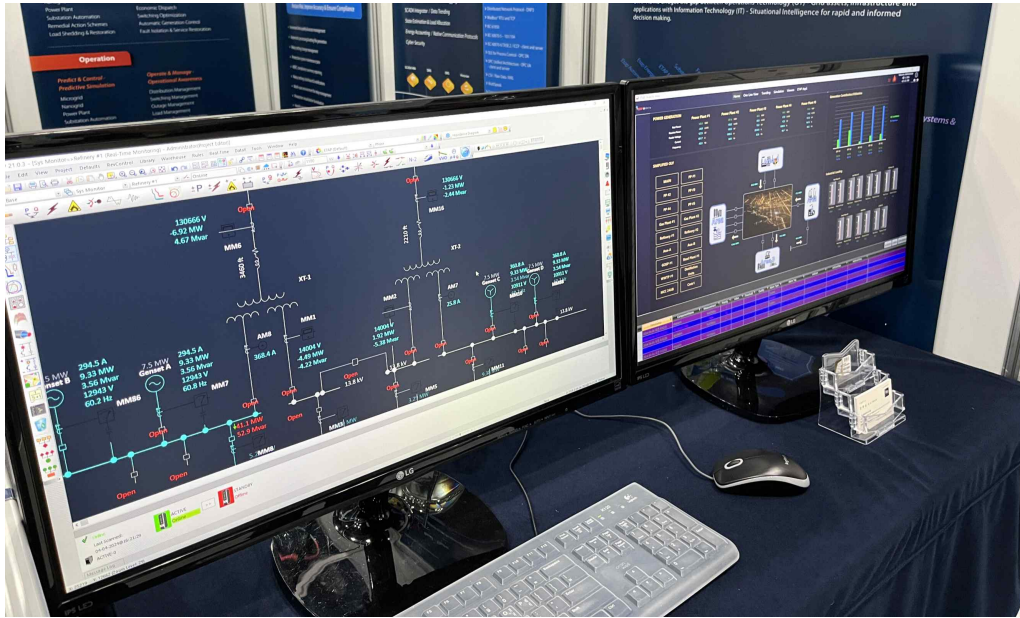


Figure 1. ETAP 전력조류 계산 프로그램 (KOREA SMART GRID EXPO 2024)

전력은 임피던스가 가장 작은 경로로 이동하며 외부로부터 교란이 생기면 빛의 속도로 급속히 전파되어 파급 고장이 원인이 되는 등 통제하기 어렵다는 특징이 있다. 하지만 이런 악조건에도 불구하고 안정적인 전력 공급을 위해서 전압 및 주파수의 심각한 요동이 허용되지 않는다. 전력 공급의 신뢰성을 방해하는 요소에 대응하기 위해서 전력이 어디로 얼마만큼 흐르고 있는지 실시간으로 계산하여 파악할 필요가 있다.

전기를 생산하고 전달하여 소비자가 사용하기까지 모든 과정에 동원되는 시스템을 전력 계통(Power System)이라고 하며 이는 발전계통, 송·변전계통, 배전계통으로 구성되어 있다. 전력 계통에서 전력이 어떻게 흐르는지에 대해 계산하는 것을 전력 조류 계산(Power Flow Calculation)이라고 하며 이를 통해 전력 계통을 구성하는 각 모선의 전압, 전류, 전력의 분포와 선로에 흐르는 전력을 파악할 수 있다. 결과적으로 파악된 정보는 전력 계통 운용 및 확충 계획, 전력 계통의 사고 예방과 경제성 증대에 활용된다.

본 프로젝트에서는 다양한 비선형 연립방정식의 수치해석적인 풀이 기법 중 가장 단순한 Jacobi Method를 활용하여 전력 조류 계산을 수행하는 프로그램을 개발하는 것이 목표이며, 앞서 보인 Figure 1과 같은 전문적인 전력 조류 계산 프로그램의 기초를 직접 구현하는 것에 의의가 있다. 또한 전력 조류 계산 코드 개발 전 코딩 능력 향상을 위한 기초 수치해석 내용에 대한 알고리즘 구성을 포함하고 있다.

2) 프로젝트 진행 일정

Week	Contents	Activity	Hour
1	전력계통공학 기초 학습	통합세미나	9
2	전력조류 계산 이론 학습	통합세미나	9
3	전력조류계산 예제 연습 및 정밀 분석	통합세미나	9
4	Gauss-Elimination 알고리즘의 구현	팀별 개발세미나	9
5	역행렬 연산 알고리즘의 구현(LU분해법 등)	팀별 개발세미나	9
6	모선 종류별 P, Q 계산 알고리즘 개발	통합세미나	9
7~8	Jacobi Method 알고리즘의 구현	팀별 개발세미나	18
9	중간 결과 보고	중간발표회 : 중간고사	9
10	개발 알고리즘의 핵심 기능 모듈의 통합	통합세미나	9
11	전공교재 예제 문제 풀이를 통한 알고리즘 성능 검증 및 비교	팀별 개발세미나	9
12~13	임의의 n모선 대응을 위한 코드 개선	팀별 개발세미나	18
14	IEEE 9모선 / 14모선 계통 전력조류 계산 풀이	팀별 개발세미나	9
15~16	결과 보고를 위한 자료 준비 / 최종 결과 보고	통합세미나 (최종 발표회 : 기말고사)	18

Figure 2. 프로젝트 진행 일정

프로젝트를 진행하기 위해 앞서 전력 계통공학 이론에 대해 학습하고 약 1달간 다양한 수치해석 기법에 대한 코딩을 진행하며 MATLAB 개발 환경에 익숙해지는 시간을 가졌다. 이후 본격적인 전력 조류 계산을 위한 코드를 개발하여 다양한 환경에서 테스트를 진행하고 올바른 결과가 나왔는지 비교 및 분석하였다.

3) 자기설계학점 행정 사항

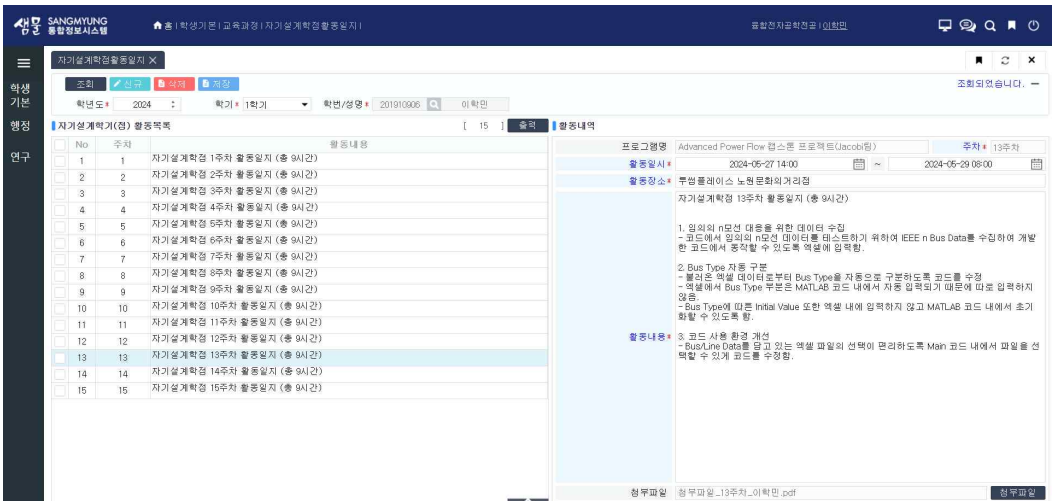


Figure 3. 자기설계학점 활동일지 (샘물 통합정보시스템 화면 캡처)

신청 학점은 9학점으로 전기공학전공 202110999 박수민 학생과 함께 팀을 이루어 일주일에 9시간 이상 활동을 진행하였으며, 활동한 내용에 대하여 샘플 상명대학교 포털 시스템에 매주 기록하였다.

II. Body

1) Matrix Calculation 모듈 개발

(1) Gauss Elimination

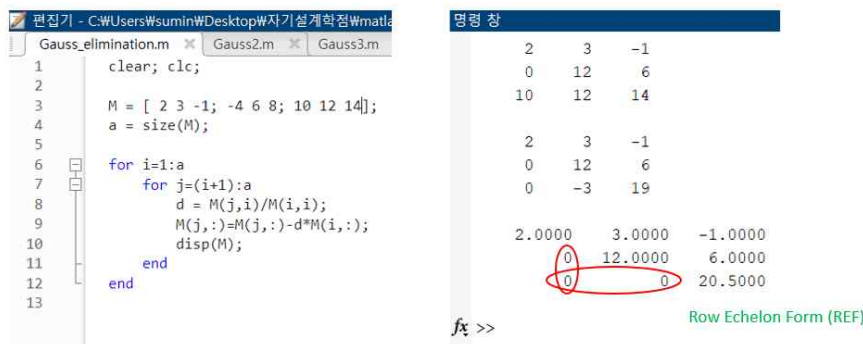
$$x_N = \frac{y_N}{A_{NN}} \quad x_k = \frac{y_k - \sum_{n=k+1}^N A_{kn}x_n}{A_{kk}} \quad k = N, N-1, \dots, 1$$

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \rightarrow \begin{bmatrix} A_{11} & A_{12} \dots & A_{1N} \\ 0 & A_{22} \dots & A_{2N} \\ \vdots & \vdots & \vdots \\ 0 & 0 \dots & A_{N-1, N-1} \\ 0 & 0 \dots 0 & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}$$

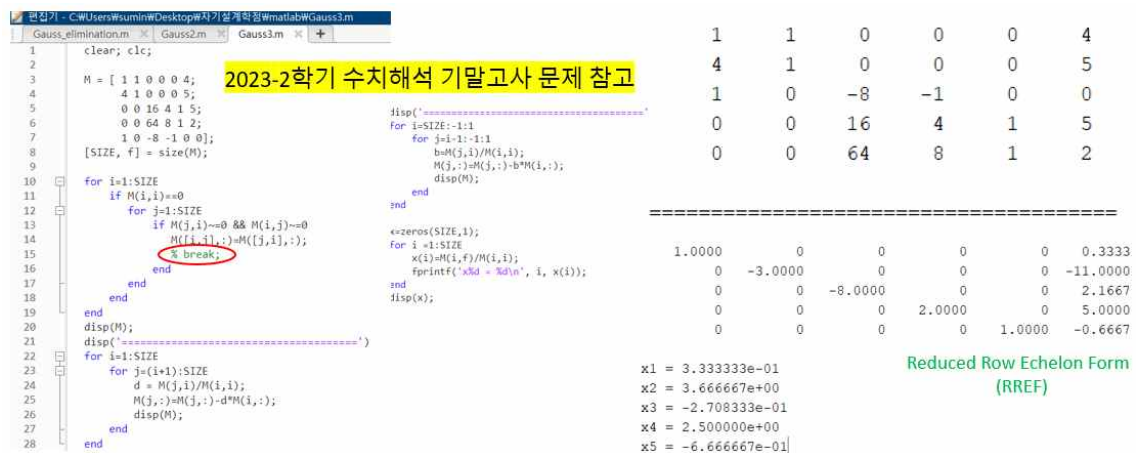
Figure 4. Gauss Elimination

가우스 소거법 (Gauss Elimination)을 정리하면 다음과 같다.

① Row Echelon Form (REF)



② Reduced Row Echelon Form (RREF)



(2) LU Decomposition

① 기본 코드

```
편집기 - C:\Users\hakmi\OneDrive\바탕 화면\학민\2024\4-1\공학\자기설계학점\코드\Gauss Elimination\GE_Level_6.m
GE_Level_6.m  GE_Level_7.m  +
63 for i = 1:rows
64     for j = (i+1):rows
65         %% Partial Pivoting 적용 시 L Matrix의 행 순서가 맞지 않게 되는 문제 발생.
66         % if abs(M(j,i)) > abs(M(i,i)) % Partial Pivoting
67         %     M([i, j], :) = M([j, i], :);
68         %     fprintf('<Partial Pivoting 수행 후 행렬, %d행 <-> %d행 교환>\n', i, j);
69         %     disp(M);
70         %     fprintf('\n\n');
71         % end
72
73         if M(j,i) ~= 0
74             fprintf('%d행 - (%d행 x %d) 연산 수행\n', j, i, M(j,i))
75             a = M(j,i) / M(i,i);
76             M(j,:) = M(j,:) - a * M(i,:);
77             L_Mat(j,i) = a;
78
79             cnt = cnt + 1;
80             fprintf('<%d iteration>\n', cnt);
81             disp(M); % 실시간 행렬 변화 출력
82             fprintf('\n');
83         end
84     end
85 end
86
87 for i = 1:rows
88     U_Mat(:,i) = M(:,i);
89 end
90
```

```
편집기 - C:\Users\hakmi\OneDrive\바탕 화면\학민\2024\4-1\공학\자기설계학점\코드\Gauss Elimination\GE_Level_6.m
GE_Level_6.m  GE_Level_7.m  +
98 % Solution
99 d = zeros(rows,1);
100 x = zeros(rows,1);
101
102 d(:,1) = b_Mat(:,1);
103
104 for i = 1:rows
105     for j = (i-1):-1:1
106         if j < 1
107             break;
108         end
109         d(i,1) = d(i,1) - d(j,1) * L_Mat(i,j);
110     end
111 end
112
113 x(:,1) = d(:,1);
114
115 for i = rows:-1:1
116     for j = rows:-1:2
117         if j == i
118             break;
119         end
120         x(i,1) = x(i,1) - U_Mat(i,j) * x(j,1);
121     end
122     x(i,1) = x(i,1) / U_Mat(i,i);
123 end
124
125 disp('해는 다음과 같습니다. ');
126
127 % Inverse Matrix Calculation
128 for k = 1:rows
129     for i = 1:rows
130         for j = (i-1):-1:1
131             if j < 1
132                 break;
133             end
134             tmp_Mat(i,k) = tmp_Mat(i,k) - tmp_Mat(j,k) * L_Mat(i,j);
135         end
136     end
137     Inv_Mat(:,k) = tmp_Mat(:,k);
138
139     for i = rows:-1:1
140         for j = rows:-1:2
141             if j == i
142                 break;
143             end
144             Inv_Mat(i,k) = Inv_Mat(i,k) - U_Mat(i,j) * Inv_Mat(j,k);
145         end
146         Inv_Mat(i,k) = Inv_Mat(i,k) / U_Mat(i,i);
147     end
148 end
149
150 fprintf('<Inverse Matrix>\n');
151 disp(Inv_Mat);
152
153 disp('<역행렬 알고리즘 점검 (내장함수 inv) 사용>');
154 disp(inv(Check_Mat));
```

Gauss Elimination을 통해 L Matrix와 U Matrix를 구하고 각 행렬을 이용하여 해와 역행렬을 계산하는 과정이다.

② Permutation Matrix

바뀐 식 순서에 대한 정보의 저장을 담당하는 행렬

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Initial Value of P Matrix

$$PA = LU$$

Original Input Matrix
Pivoted Matrix

$$(PA)^{-1} = (LU)^{-1}$$

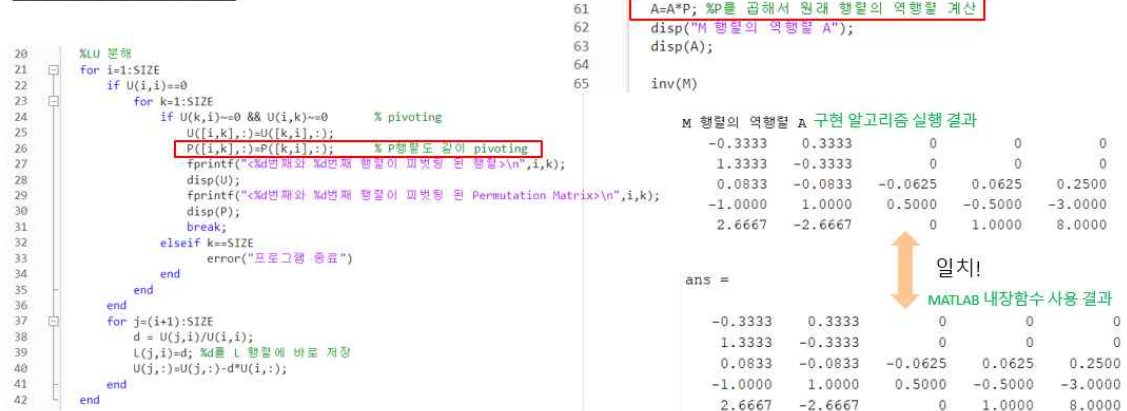
$$A^{-1}P^{-1} = (LU)^{-1}$$

$$A^{-1} = (LU)^{-1}P$$

Figure 5. Permutation Matrix

Gauss Elimination을 수행하는 과정에서 대각 원소에 0이 위치하지 않도록 식의 순서를 바꾸게 되는데, Permutation Matrix를 통하여 바뀐 식 순서에 대한 정보를 저장하였다.

Permutation Matrix



앞서 유도한 공식을 추가 적용하였고 MATLAB의 역행렬 계산 내장 함수 결과와 비교하여 역행렬 알고리즘 구성이 올바르게 이루어졌음을 확인하였다.

행렬 계산 관련(Gauss Elimination, LU Decomposition, Inverse Matrix, etc.) 코드는 다음 GitHub 링크에서 확인할 수 있다.

https://github.com/hakmin1015/AdvancedPowerFlow_CapstoneProject/tree/main/Matrix_Calculation

2) 선형방정식의 Iterative Solution 개발 및 성능 비교

Jacobi Method

The Jacobi method uses the "old" values of $x(i)$ at iteration i on the right side of (6.2.4) to generate the "new" value $x_k(i+1)$ on the left side of (6.2.4). That is,

$$x_k(i+1) = \frac{1}{A_{kk}} \left[y_k - \sum_{s=1}^{i-1} A_{ks}x_s(i) - \sum_{s=i+1}^N A_{ks}x_s(i) \right] \quad k = 1, 2, \dots, N \quad (6.2.5)$$

The Jacobi method given by (6.2.5) also can be written in the following matrix format:

$$x(i+1) = Mx(i) + D^{-1}y \quad (6.2.6)$$

where

$$M = D^{-1}(D - A) \quad (6.2.7)$$

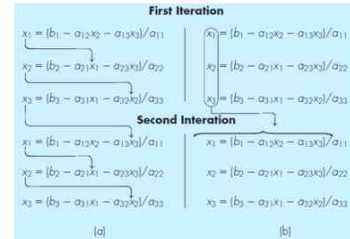
and

$$D = \begin{bmatrix} A_{11} & 0 & 0 & \dots & 0 \\ 0 & A_{22} & 0 & \dots & 0 \\ 0 & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_{NN} \end{bmatrix} \quad (6.2.8)$$

Gauss-Seidel Method

The Gauss-Seidel method is given by

$$x_k(i+1) = \frac{1}{A_{kk}} \left[y_k - \sum_{s=1}^{k-1} A_{ks}x_s(i+1) - \sum_{s=k+1}^N A_{ks}x_s(i) \right] \quad (6.2.9)$$



(a) the Gauss-Seidel and (b) the Jacobi iterative methods

- Method별 수행 시간 측정 : MATLAB의 tic 및 toc 함수 활용
- Jacobi Method에 비해 Gauss-Seidel Method의 성능이 더 우수한지 확인

Figure 6. Iterative Solution to Linear Algebraic Equations

선형방정식을 풀기 위한 수치해석적 방법은 대표적으로 Jacobi Method와 Gauss-Seidel Method가 있다. Gauss-Seidel Method는 Jacobi Method에서 수렴성을 개선한 방법으로, 실제 코드를 구현하여 두 방법 간의 성능 차이를 비교 및 분석하였다.

(1) Jacobi Method

```
Jacobi_Solution.m  Gauss_Seidel_Solution.m  Jacobi_Method.m  +
1 % Jacobi Method : iterative solution to linear algebraic equations
2
3
4 function Jacobi_Method(D_Inv, D, A)
5
6 [rows, ~] = size(A);
7 AA = zeros(rows); % not Augmented Matrix
8
9 for i = 1:rows
10     AA(:,i) = A(:,i);
11 end
12
13 M = D_Inv*(D-AA); % M = D^(-1)*(D-A)
14
15 i = 1;
16 lmt = 30; % Iteration Limit
17
18 x_Mat = zeros(rows,lmt);
19 y_Mat = A(:,end);
20 err = zeros(rows,1);
21
22 while true
23     x_Mat(:,i+1) = M*x_Mat(:,i) + D_Inv*y_Mat;
24
25     for k = 1:rows
26         err(k) = abs((x_Mat(k,i+1)-x_Mat(k,i))/x_Mat(k,i+1)); % 근사 상대오차
27     end
28
29     i = i+1;
30
31     if i>lmt || sum(err)>0.0001 == 0
32         break;
33     end
34 end
35
36 fprintf('=====Jacobi Solution=====\n\n')
37
38 for k = 1:i
39     fprintf('Iteration %d\n', k-1);
40     for m = 1:rows
41         fprintf('x%d(%d) = %f\n', m, k-1, x_Mat(m,k));
42     end
43     fprintf('\n');
44 end
45
46 fprintf('Final Approximate Percent Relative Error\n\n');
47 for k = 1:rows
48     fprintf('err(%d) = %f%%\n', k, err(k)*100);
49 end
50
51 end
```


Jacobi Method의 Iterative Limit을 30으로 설정하고, 근사 상대 오차가 30번의 반복 동안 목표한 값 이내로 수렴하지 못하면 발산한 것으로 간주하고 프로그램을 강제 종료한다. 설정한 근사 상대 오차는 0.0001 즉 0.01[%]이다.

(2) Gauss-Seidel Method

```

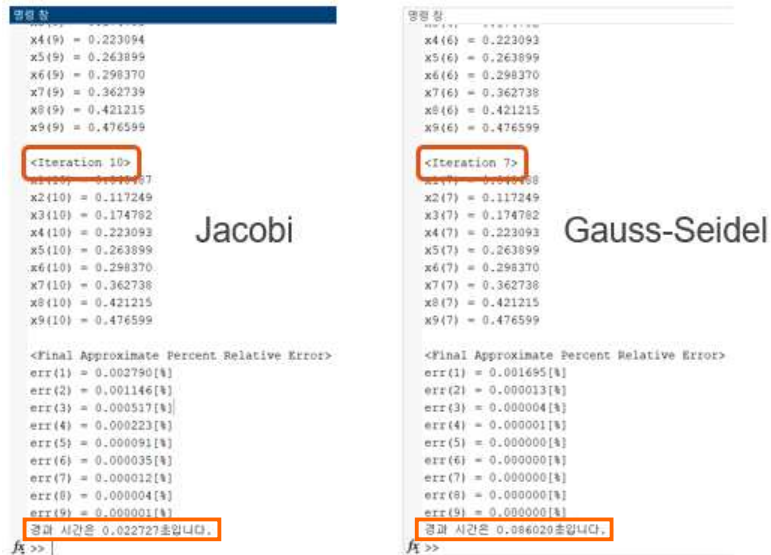
1 % Gauss-Seidel Method : iterative solution to linear algebraic equations(...)
3
4 function Gauss_Seidel_Method(D_Inv, D, A)
5
6     [rows, ~] = size(A);
7     AA = zeros(rows); % not Augmented Matrix
8
9     for i = 1:rows
10        AA(:,i) = A(:,i);
11    end
12
13    M = D_Inv*(D-AA); % M = D^(-1)*(D-A)
14
15    i = 1;
16    lmt = 30; % Iteration Limit
17
18    x_Mat = zeros(rows,lmt);
19    y_Mat = A(:,end);
20    err = zeros(rows,1);
21
22    while true
23        x_Mat(:,i+1) = M*x_Mat(:,i) + D_Inv*y_Mat;
24
25        for k = 1:rows
26            err(k) = abs((x_Mat(k,i+1)-x_Mat(k,i))/x_Mat(k,i+1)); % 근사 상대오차
27        end
28
29        i = i+1;
30
31        if i>lmt || sum(err)>0.0001 == 0
32            break;
33        end
34    end
35
36    fprintf('=====Gauss-Seidel Solution=====\n\n')
37
38    for k = 1:i
39        fprintf('Iteration %d\n', k-1);
40        for m = 1:rows
41            fprintf('x%d(%d) = %f\n', m, k-1, x_Mat(m,k));
42        end
43        fprintf('\n');
44    end
45
46    fprintf('<Final Approximate Percent Relative Error>\n');
47    for k = 1:rows
48        fprintf('err(%d) = %f[%%]\n', k, err(k)*100);
49    end
50
51 end

```

기본적인 코드의 구성은 Jacobi Method와 동일하지만, 함수의 입력값으로 들어오는 D_Inv 변수의 차이로 Gauss-Seidel Method가 구현된다.

(3) Jacobi vs Gauss-Seidel

① MATLAB 코드 실행 결과



Gauss-Seidel은 Jacobi에 비해 LU Decomposition 과정이 복잡하나, 총 Iteration 수가 적다. H/W의 처리 속도가 매우 빨라 두 Method 간의 코드 실행 속도 차이가 유의미하게 나타나지 않았으나, 구하고자 하는 변수의 개수가 많아질수록 코드 실행 시간에 점차 차이가 생길 것으로 예상된다.

② EXCEL을 활용한 수렴성 비교 그래프

Real Value	
x1	0.048487553
x2	0.117248918
x3	0.174782442
x4	0.223093109
x5	0.263898793
x6	0.298369877
x7	0.362738472
x8	0.421215434
x9	0.476599143

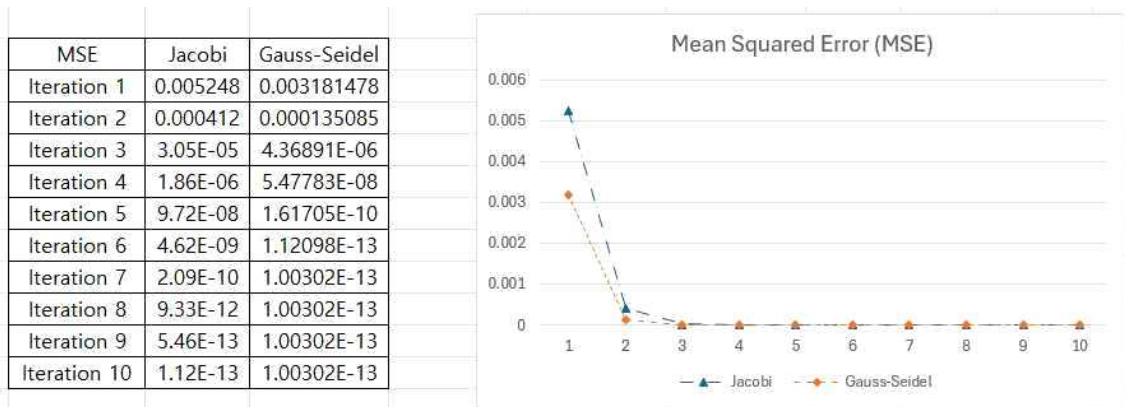
MSE

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$$

입력한 행렬의 실제 해와 오차를 계산하기 위한 MSE(Mean Squared Error) 수식은 위와 같다.

Jacobi										Gauss-Seidel									
Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8	Iteration 9	Iteration 10	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8	Iteration 9	Iteration 10
0.1	0.026049	0.056776	0.046212	0.04904	0.04837	0.04851	0.04848	0.04849	0.048487	0.1	0.03021	0.05325	0.04784	0.04853	0.04849	0.04849	0.04849	0.04849	0.048488
0.181818	0.08956	0.126322	0.114866	0.11781	0.11713	0.11728	0.11724	0.11725	0.117249	0.172727	0.10074	0.12056	0.11699	0.11726	0.11725	0.11725	0.11725	0.11725	0.117249
0.25	0.146112	0.182485	0.172944	0.17519	0.1747	0.1748	0.17478	0.17478	0.174782	0.235606	0.15797	0.1769	0.17469	0.17478	0.17478	0.17478	0.17478	0.17478	0.174782
0.307692	0.196566	0.229077	0.221844	0.22335	0.22304	0.2231	0.22309	0.22309	0.223093	0.289569	0.20878	0.22413	0.22307	0.22309	0.22309	0.22309	0.22309	0.22309	0.223093
0.357143	0.24173	0.268131	0.263106	0.26405	0.26387	0.2639	0.2639	0.2639	0.263899	0.336459	0.25382	0.26421	0.2639	0.2639	0.2639	0.2639	0.2639	0.2639	0.263899
0.4	0.282318	0.301142	0.29791	0.29845	0.29836	0.29837	0.29837	0.29837	0.29837	0.377569	0.29406	0.29842	0.29837	0.29837	0.29837	0.29837	0.29837	0.29837	0.29837
0.4375	0.351838	0.364274	0.362495	0.36278	0.36273	0.36274	0.36274	0.36274	0.362738	0.413902	0.36153	0.36273	0.36274	0.36274	0.36274	0.36274	0.36274	0.36274	0.362738
0.470588	0.415441	0.422018	0.421106	0.42123	0.42121	0.42122	0.42122	0.42122	0.421215	0.446241	0.42137	0.42122	0.42122	0.42122	0.42122	0.42122	0.42122	0.42122	0.421215
0.5	0.473856	0.47692	0.476555	0.47661	0.4766	0.4766	0.4766	0.4766	0.476599	0.475209	0.47659	0.4766	0.4766	0.4766	0.4766	0.4766	0.4766	0.4766	0.476599
(y-yhat)^2	0.002653532	0.000503489	6.86984E-05	5.17814E-06	3E-07	1.5E-08	7E-10	3.1E-11	2.1E-12	3.059E-13	0.002653532	0.00033	2.3E-05	4.2E-07	1.4E-09	3.1E-13	2E-13	2E-13	1.997E-13
	0.004169166	0.000766676	8.23208E-05	5.6783E-06	3.1E-07	1.5E-08	6.8E-10	3.5E-11	1.2E-12	6.653E-15	0.003077818	0.00027	1.1E-05	6.5E-08	5E-11	6.7E-15	6.7E-15	6.7E-15	6.653E-15
	0.005657681	0.000821994	5.93294E-05	3.7987E-06	1.7E-07	7.5E-09	3.4E-10	1.2E-11	3.1E-13	1.952E-13	0.003699505	0.00028	4.5E-06	9.1E-09	2.4E-12	2E-13	2E-13	2E-13	1.952E-13
	0.007156972	0.000703688	3.5807E-05	1.56027E-06	6.6E-08	2.7E-09	1.2E-10	4.4E-12	7.9E-13	1.186E-14	0.004419044	0.0002	1.1E-06	3.7E-10	1.2E-14	1.2E-14	1.2E-14	1.2E-14	1.186E-14
	0.008694482	0.000491455	1.79116E-05	6.28521E-07	2.2E-08	7.7E-10	2.7E-11	6.3E-13	4.3E-14	4.268E-14	0.005264984	0.0001	9.6E-08	3.2E-12	4.3E-14	4.3E-14	4.3E-14	4.3E-14	4.268E-14
	0.010328682	0.000257663	7.68466E-06	2.11487E-07	6.3E-09	1.9E-10	4.5E-12	7.7E-13	1.5E-14	1.508E-14	0.006272501	1.9E-05	2.5E-09	1.3E-12	1.5E-14	1.5E-14	1.5E-14	1.5E-14	1.508E-14
	0.005589286	0.00011882	2.35785E-06	5.92784E-08	1.5E-09	4.2E-11	2.3E-12	2.2E-13	2.8E-13	2.224E-13	0.002617707	1.5E-06	1.6E-10	2.2E-13	2.2E-13	2.2E-13	2.2E-13	2.2E-13	2.224E-13
	0.00243765	3.33441E-05	6.44111E-07	1.19759E-08	2.7E-10	5.9E-12	3.2E-13	1.9E-13	1.9E-13	1.888E-13	0.000626279	2.3E-08	2.5E-12	1.9E-13	1.9E-13	1.9E-13	1.9E-13	1.9E-13	1.888E-13
	0.0005476	7.52483E-06	1.0295E-07	1.94856E-09	3.4E-11	1.3E-12	2E-14	2E-14	2E-14	2.031E-14	1.9325E-06	6.6E-11	2E-14	2E-14	2E-14	2E-14	2E-14	2E-14	2.031E-14
MSE	0.005248339	0.000411628	3.05396E-05	1.85664E-06	9.7E-08	4.6E-09	2.1E-10	9.3E-12	5.5E-13	1.121E-13	0.003181478	0.00014	4.4E-06	5.5E-08	1.6E-10	1.1E-13	1E-13	1E-13	1.003E-13

각 Iteration 결과를 엑셀에 정리하고 $(y - \hat{y})^2$ 을 계산하여 최종적으로 MSE를 계산하였다.



최종 결과를 표로 정리하고 그래프로 나타낸 결과이다. Jacobi Method에 비해 Gauss-Seidel Method의 수렴 속도가 더 빠름을 확인할 수 있다. 그러나 Jacobi 반복의 개념이 더 간단하고 병렬성을 추출하기가 용이하며 같은 반복 주기 동안에 구하려는 미지의 값이 변하지 않기 때문에 통신 부하를 크게 줄일 수 있다는 장점이 있다.¹⁾

Iterative Solution (Jacobi & Gauss-Seidel Method)에 대한 코드는 다음 GitHub 링크에서 확인할 수 있다.

https://github.com/hakmin1015/AdvancedPowerFlow_CapstoneProject/tree/main/Iterative_Solutions

1) 채수환, 이용두. (1993). MIMD 컴퓨터에서 선형시스템을 풀기 위한 반복병렬알고리즘. (구)정보과학회 논문지, 20(6), 802-809.

$$V_1(1) = \frac{1}{Y_{11}} \left\{ \frac{P_1 - jQ_1}{V_1^2(1)} - \sum_{k=2}^5 Y_{1k} V_k(1) + Y_{10} V_0(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ \frac{-4.4 - j2.8}{1.999 \angle 90.000^\circ} - Y_{12} V_2^0 - Y_{13} V_3^0 - Y_{14} V_4^0 - Y_{15} V_5^0 \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ \frac{-4.4 - j2.8}{1.999 \angle 90.000^\circ} - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) \right\}$$

$$= 0.8246 - j0.2599 = 0.8623 \angle -16.995^\circ \quad [p.u.]$$

[Revised]

$$V_2(1) = 0.8599 \angle -16.995^\circ = 0.8599 \angle -16.995^\circ \quad [p.u.]$$

$$V_3(1) = \frac{1}{Y_{33}} \left\{ \frac{P_3 - jQ_3}{V_3^2(1)} - \sum_{k=1,2,4,5} Y_{3k} V_k(1) + Y_{30} V_0(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ \frac{4.4 - j2.8}{1.999 \angle 90.000^\circ} - Y_{31} V_1(1) - Y_{32} V_2(1) - Y_{34} V_4(1) - Y_{35} V_5(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ \frac{4.4 - j2.8}{1.999 \angle 90.000^\circ} - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) - (1.999 \angle 90.000^\circ) (1.0 \angle 0^\circ) \right\}$$

$$= 1.0530 + j0.0500 = 1.0534 \angle 2.700^\circ \quad [p.u.]$$

[Revised]

$$V_5(1) = 1.0497 + j0.0492 = 1.0490 \angle 2.7230^\circ \quad [p.u.]$$

$$V_4(1) = \frac{1}{Y_{44}} \left\{ \frac{P_4 - jQ_4}{V_4^2(1)} - \sum_{k=1,2,3,5} Y_{4k} V_k(1) + Y_{40} V_0(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ 0 - Y_{41} V_1(1) - Y_{42} V_2(1) - Y_{43} V_3(1) - Y_{45} V_5(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ 0 - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (1.0534 \angle 2.700^\circ) - (1.999 \angle 90.000^\circ) (1.0490 \angle 2.7230^\circ) \right\}$$

$$= 1.0432 - j0.2093 = 1.0432 \angle -11.0169^\circ \quad [p.u.]$$

[Revised]

$$V_4(1) = 1.0432 \angle -11.0169^\circ \quad [p.u.]$$

$$V_5(1) = \frac{1}{Y_{55}} \left\{ \frac{P_5 - jQ_5}{V_5^2(1)} - \sum_{k=1,2,3,4} Y_{5k} V_k(1) + Y_{50} V_0(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ 0 - Y_{51} V_1(1) - Y_{52} V_2(1) - Y_{53} V_3(1) - Y_{54} V_4(1) \right\}$$

$$= \frac{1}{j1.999 \angle 90.000^\circ} \left\{ 0 - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (0.8599 \angle -16.995^\circ) - (1.999 \angle 90.000^\circ) (1.0534 \angle 2.700^\circ) - (1.999 \angle 90.000^\circ) (1.0432 \angle -11.0169^\circ) \right\}$$

$$= 0.9863 - j0.0493 = 0.9862 \angle -2.7082^\circ \quad [p.u.]$$

[Revised]

$$V_5(1) = 0.9862 \angle -2.7082^\circ \quad [p.u.]$$

$$P_1 = \sum_{k=1}^5 Y_{1k} V_1(1) \cos(\delta_1(1) - \delta_k(1) + \theta_{1k})$$

$$= Y_{11} V_1(1) \cos(\delta_1(1) + \theta_{11}) + Y_{12} V_2(1) \cos(\delta_1(1) - \delta_2(1) + \theta_{12})$$

$$= 4.4 \cos(0 + 0.999 \angle 90^\circ) + 1.999 \times 1.0534 \cos(0 - 16.995^\circ + 90.000^\circ)$$

$$= -11.0169 \quad [p.u.]$$

$$Q_1 = -\sum_{k=1}^5 Y_{1k} V_1(1) \sin(\delta_1(1) - \delta_k(1) + \theta_{1k})$$

$$= -\left[Y_{11} V_1(1) \sin(\delta_1(1) + \theta_{11}) + Y_{12} V_2(1) \sin(\delta_1(1) - \delta_2(1) + \theta_{12}) \right]$$

$$= -\left[1.999 \times 1.0534 \sin(0 - 16.995^\circ + 90.000^\circ) + 4.4 \cos(0 + 0.999 \angle 90^\circ) \right]$$

$$= -6.8219 \quad [p.u.]$$

Iteration Results (Jacobi Method Used) [p.u.]

Initial Value		
$V_1(0) = 1.0 \angle 0^\circ$	$P_1 = ?$	$Q_1 = ?$
$V_2(0) = 1.0 \angle 0^\circ$	$P_2 = -8.0$	$Q_2 = -2.8$
$V_3(0) = 1.05 \angle 0^\circ$	$P_3 = 4.4$	$Q_3 = ?$
$V_4(0) = 1.0 \angle 0^\circ$	$P_4 = 0$	$Q_4 = 0$
$V_5(0) = 1.0 \angle 0^\circ$	$P_5 = 0$	$Q_5 = 0$

Iteration 1		
$V_1(1) = 1.0 \angle 0^\circ$	$P_1 = 0$	$Q_1 = 0$
$V_2(1) = 0.8946 \angle -15.6960^\circ$	$P_2 = -8.0$	$Q_2 = -2.8$
$V_3(1) = 1.0409 \angle 2.8509^\circ$	$P_3 = 4.4$	$Q_3 = 4.0$
$V_4(1) = 1.0408 \angle -0.0424^\circ$	$P_4 = 0$	$Q_4 = 0$
$V_5(1) = 1.0060 \angle -0.0288^\circ$	$P_5 = 0$	$Q_5 = 0$

Bus 3 Type: PV Bus \rightarrow PQ Bus

Iteration 2		
$V_1(2) = 1.0 \angle 0^\circ$	$P_1 = 0.0028$	$Q_1 = -0.3002$
$V_2(2) = 0.8990 \angle -16.8995^\circ$	$P_2 = -8.0$	$Q_2 = -2.8$
$V_3(2) = 1.0801 \angle 2.0498^\circ$	$P_3 = 4.4$	$Q_3 = 4.0$
$V_4(2) = 1.0256 \angle 0.5952^\circ$	$P_4 = 0$	$Q_4 = 0$
$V_5(2) = 0.9933 \angle -2.5932^\circ$	$P_5 = 0$	$Q_5 = 0$

Iteration 3		
$V_1(3) = 1.0 \angle 0^\circ$	$P_1 = -11.0169$	$Q_1 = -6.8219$
$V_2(3) = 0.8599 \angle -16.9983^\circ$	$P_2 = -8.0$	$Q_2 = -2.8$
$V_3(3) = 1.0690 \angle 2.5230^\circ$	$P_3 = 4.4$	$Q_3 = 4.0$
$V_4(3) = 1.0482 \angle -0.2159^\circ$	$P_4 = 0$	$Q_4 = 0$
$V_5(3) = 0.9892 \angle -2.5082^\circ$	$P_5 = 0$	$Q_5 = 0$

Jacobi Method를 이용하여 3회 반복 계산한 과정과 결과는 다음과 같다. 손으로 계산하기에 연산량이 방대하여 컴퓨터 프로그램의 개발이 불가피함을 알 수 있다.

(2) 코드 개발

① Y Bus Matrix

- Node Voltage Method based on KCL
- Network Equation : $YV = I$
- Y is called "Y bus Matrix"
- I : Bus로 주입되는 전류
- $N \times N$ bus admittance matrix (symmetric)
 - Diagonal elements
 $Y_{kk} = \text{sum of } Y \text{ connected to bus } k \text{ (} k = 1, 2, \dots, N \text{)}$
 - Off-diagonal elements
 $Y_{kn} = -(\text{sum of } Y \text{ connected between buses } k \text{ and } n) \text{ (} k \neq n \text{)}$

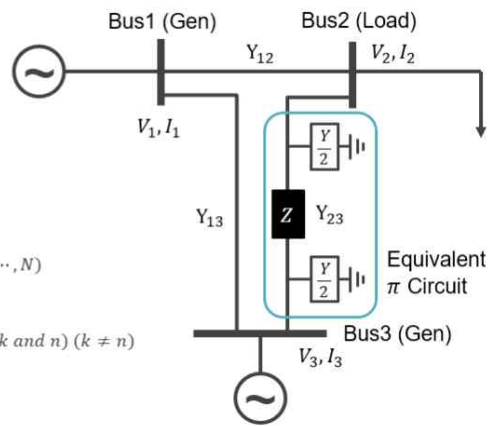


Figure 7. Y Bus Matrix Calculation

코드)

```

1 % Y matrix Calculation
2
3 function Y = Y_Mat_Calc(SIZE,L_data)
4
5     Y = zeros(SIZE);
6
7     for i = 1:SIZE % Diagonal elements
8         for j = 1:SIZE
9             if i==j
10                for k = 1:SIZE
11                    if complex(L_data(i,1,k),L_data(i,2,k)) ~= 0
12                        Y(i,j) = Y(i,j) + 1/complex(L_data(i,1,k),L_data(i,2,k));
13                    end
14                    Y(i,j) = Y(i,j) + complex(L_data(i,3,k),L_data(i,4,k))/2;
15                end
16            else % Off-diagonal elements
17                if complex(L_data(i,1,j),L_data(i,2,j)) ~= 0
18                    Y(i,j) = -(1/complex(L_data(i,1,j),L_data(i,2,j)));
19                else
20                    Y(i,j) = 0;
21                end
22            end
23        end
24    end
25 end
26

```

실행 결과 예시)

	1	2	3	4	5
1	3.7290 - 49.7203i	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	-3.7290 + 49.7203i
2	0.0000 + 0.0000i	2.6783 - 28.4590i	0.0000 + 0.0000i	-0.8928 + 9.9197i	-1.7855 + 19.8393i
3	0.0000 + 0.0000i	0.0000 + 0.0000i	7.4580 - 99.4406i	-7.4580 + 99.4406i	0.0000 + 0.0000i
4	0.0000 + 0.0000i	-0.8928 + 9.9197i	-7.4580 + 99.4406i	1.1922e+01 - 1.4796e+02i	-3.5711 + 39.6786i
5	-3.7290 + 49.7203i	-1.7855 + 19.8393i	0.0000 + 0.0000i	-3.5711 + 39.6786i	9.0856e+00 - 1.0858e+02i

② PQ (Load) Bus Calculation

- For Load Bus(PQ Bus)

$$YV = I$$

- To compute V_k and δ_k

$$V_k(i+1) = V_k \angle \delta_k(i+1)$$

$$S_k = V_k I_k^* \rightarrow I_k^* = \frac{S_k}{V_k} = \frac{P_k + jQ_k}{V_k} \rightarrow I_k = \frac{P_k - jQ_k}{V_k^*} = \frac{1}{Y_{kk}} \left[\frac{P_k - jQ_k}{V_k^*(i)} - \sum_{n=1}^{k-1} Y_{kn} V_n \angle \delta_n(i) - \sum_{n=k+1}^N Y_{kn} V_n \angle \delta_n(i) \right]$$

Figure 8. PQ Bus Calculation

코드)

```

1 % PQ(Load) Bus Calculation
2
3 function [V,Delta] = PQ_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q)
4
5     I_k = (P(k,i)-sqrt(-1)*Q(k,i)) / (V(k,i)*exp(-sqrt(-1)*Delta(k,i)*(pi/180)));
6
7     Sum_YV_1 = 0;
8     Sum_YV_2 = 0;
9
10    for n = 1:k-1
11        Sum_YV_1 = Sum_YV_1 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
12    end
13
14    for n = k+1:SIZE
15        Sum_YV_2 = Sum_YV_2 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
16    end
17
18    V_k = (1/Y(k,k)) * (I_k - Sum_YV_1 - Sum_YV_2); % k모선의 i번째 iteration 값 계산
19    V(k,i+1) = sqrt(power(real(V_k),2) + power(imag(V_k),2));
20    Delta(k,i+1) = atan(imag(V_k) / real(V_k)) * (180/pi);
21
22    % Recalculation
23    I_k = (P(k,i)-sqrt(-1)*Q(k,i)) / (V(k,i+1)*exp(-sqrt(-1)*Delta(k,i+1)*(pi/180)));
24    V_k = (1/Y(k,k)) * (I_k - Sum_YV_1 - Sum_YV_2);
25    V(k,i+1) = sqrt(power(real(V_k),2) + power(imag(V_k),2));
26    Delta(k,i+1) = atan(imag(V_k) / real(V_k)) * (180/pi);

```


③ PV (Voltage-Controlled) Bus Calculation

- For Voltage-controlled Bus(PV Bus)

- To computes Q_k and δ_k

$$YV = I$$

- ✓ If Q_G violates its limit, cannot maintain voltage

$$V_k(i+1) = V_k \angle \delta_k(i+1)$$

$$Q_k = V_k(i) \sum_{n=1}^N Y_{kn} V_n(i) \sin[\delta_k(i) - \delta_n(i) - \theta_{kn}]$$

$$= \frac{1}{Y_{kk}} \left[\frac{P_k - jQ_k}{V_k(i)} - \sum_{n=1}^{k-1} Y_{kn} V_n(i) \angle \delta_n(i) - \sum_{n=k+1}^N Y_{kn} V_n(i) \right]$$

Figure 9. PV Bus Calculation

코드)

```

1 % PV(Gen) Bus Calculation
2
3 function [Delta,V] = PV_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q)
4
5     I_k = (P(k,i)-sqrt(-1)*Q(k,i+1)) / (V(k,i)*exp(-sqrt(-1)*Delta(k,i)*(pi/180)));
6
7     Sum_YV_1 = 0;
8     Sum_YV_2 = 0;
9
10    for n = 1:k-1
11        Sum_YV_1 = Sum_YV_1 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
12    end
13
14    for n = k+1:SIZE
15        Sum_YV_2 = Sum_YV_2 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
16    end
17
18    V_k = (1/Y(k,k)) * (I_k - Sum_YV_1 - Sum_YV_2); % k모선의 i번째 iteration 값 계산
19
20    Delta(k,i+1) = atan(imag(V_k) / real(V_k)) * (180/pi);
21
22    V(k,i+1) = V(k,i); % V value never change if it's PV Bus
23 end

1 % Calculate PV Bus' Q
2
3 function [Q,Q_G,Q_L] = PV_Q_Calc(k,i,Y,V,Delta,Q_G,Q_L,Q,SIZE)
4
5     Sum_YVsin = 0;
6     for n = 1:SIZE
7         Sum_YVsin = Sum_YVsin + abs(Y(k,n)) * V(n,i) ...
8             * sin((Delta(k,i) - Delta(n,i) - angle(Y(k,n))*(180/pi)) * (pi/180));
9     end
10
11     Q(k,i+1) = V(k,i) * Sum_YVsin;
12     Q_L(k,i+1) = Q_L(k,i);
13     Q_G(k,i+1) = Q(k,i+1) + Q_L(k,i+1);
14
15 end

```

④ Slack (Swing) Bus Calculation

- For Swing Bus(Slack Bus)

- No iterations are needed to compute P_1 and Q_1 for swing bus

$$P_1 = V_1 \sum_{n=1}^N Y_{1n} V_n \cos(\delta_1 - \delta_n - \theta_{1n}) = \sum_{n=1}^N Y_{1n} V_n \cos(\delta_n + \theta_{1n})$$

$$Q_1 = V_1 \sum_{n=1}^N Y_{1n} V_n \sin(\delta_1 - \delta_n - \theta_{1n}) = - \sum_{n=1}^N Y_{1n} V_n \sin(\delta_n + \theta_{1n})$$

- 최종적으로 구해진 V, δ 의 값을 대입하여 P, Q 를 계산할 수 있음.

Figure 10. Slack Bus Calculation

코드)

```

1 % Slack(Swing) Bus Calculation
2
3 function [P,Q,P_G,Q_G] = Slack_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q,P_G,Q_G,P_L,Q_L)
4
5 for n = 1:SIZE
6     P(k,i+1) = P(k,i+1) + V(k,1)*abs(Y(1,n))*V(n,i+1)*cos((Delta(n,i+1)+angle(Y(1,n))*(180/pi))*(pi/180))
7 end
8
9 for n = 1:SIZE
10    Q(k,i+1) = Q(k,i+1) + V(k,1)*abs(Y(1,n))*V(n,i+1)*sin((Delta(n,i+1)+angle(Y(1,n))*(180/pi))*(pi/180))
11 end
12 Q(k,i+1) = -Q(k,i+1);
13
14 P_G(k,i+1) = P(k,i+1) - P_L(k,i+1);
15 Q_G(k,i+1) = Q(k,i+1) - Q_L(k,i+1);
16
17 end

```

⑤ Bus Calculation Integration

```

3 function [V,Delta,P,Q,P_G,P_L,Q_G,Q_L,Bus_Type,Switch_Sig] ...
4     = Total_Bus_Calc(i,k,SIZE,Y,V,Delta,P,Q,P_G,P_L,Q_G,Q_L,Q_Gmax,Q_Gmin,Bus_Type,Switch_Sig)
5
6 switch Bus_Type(k,i)
7 case 0 % For Slack Bus (Swing)
8     V(k,i+1) = V(k,i); % V = 180 (초기 설정)을 계속 유지
9     Delta(k,i+1) = Delta(k,i);
10    P_G(k,i+1) = P_G(k,i);
11    Q_G(k,i+1) = Q_G(k,i);
12    P_L(k,i+1) = P_L(k,i);
13    Q_L(k,i+1) = Q_L(k,i);
14    P(k,i+1) = P_G(k,i+1) - P_L(k,i+1);
15    Q(k,i+1) = Q_G(k,i+1) - Q_L(k,i+1);
16
17 case 1 % For PV Bus (Gen)
18    [Q_G,Q_L] = PV_Q_Calc(k,i,Y,V,Delta,Q_G,Q_L,Q,SIZE);
19    P_G(k,i+1) = P_G(k,i);
20    P_L(k,i+1) = P_L(k,i);
21    P(k,i+1) = P_G(k,i+1) - P_L(k,i+1);
22
23    [Delta,V] = PV_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q);
24
25 case 2 % For PQ Bus (Load)
26    P_G(k,i+1) = P_G(k,i);
27    P_L(k,i+1) = P_L(k,i);
28    Q_G(k,i+1) = Q_G(k,i);
29    Q_L(k,i+1) = Q_L(k,i);
30    P(k,i+1) = P_G(k,i+1) - P_L(k,i+1);
31    Q(k,i+1) = Q_G(k,i+1) - Q_L(k,i+1);
32
33    [V,Delta] = PQ_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q);

```

⑥ Bus Type Switching

코드)

```

35         case 3 % For PV -> PQ Bus (for keep voltage value)
36             P_G(k,i+1) = P_G(k,i);
37             P_L(k,i+1) = P_L(k,i);
38             Q_L(k,i+1) = Q_L(k,i);
39             P(k,i+1) = P_G(k,i+1) - P_L(k,i+1);
40
41             if Switch_Sig(k,i) == 1
42                 Q_G(k,i) = Q_Gmin(k,1);
43                 Q(k,i) = Q_G(k,i) - Q_L(k,i);
44                 Q_G(k,i+1) = Q_G(k,i);
45                 Q(k,i+1) = Q_G(k,i+1) - Q_L(k,i+1);
46
47             elseif Switch_Sig(k,i) == 2
48                 Q_G(k,i) = Q_Gmax(k,1);
49                 Q(k,i) = Q_G(k,i) - Q_L(k,i);
50                 Q_G(k,i+1) = Q_G(k,i);
51                 Q(k,i+1) = Q_G(k,i+1) - Q_L(k,i+1);
52
53             else
54                 Q_G(k,i+1) = Q_G(k,i);
55                 Q(k,i+1) = Q_G(k,i+1) - Q_L(k,i+1);
56             end
57
58             [V,Delta] = PV2PQ_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q);
59         end

```

```

1 % PV->PQ Bus Calculation
2
3 function [V,Delta] = PV2PQ_Bus_Calc(SIZE,k,i,Y,V,Delta,P,Q)
4
5     I_k = (P(k,i)-sqrt(-1)*Q(k,i)) / (V(k,i)*exp(-sqrt(-1)*Delta(k,i)*(pi/180)));
6
7     Sum_YV_1 = 0;
8     Sum_YV_2 = 0;
9
10 for n = 1:k-1
11     Sum_YV_1 = Sum_YV_1 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
12 end
13
14 for n = k+1:SIZE
15     Sum_YV_2 = Sum_YV_2 + Y(k,n) * (V(n,i)*exp(sqrt(-1)*Delta(n,i)*(pi/180)));
16 end
17
18 V_k = (1/Y(k,k)) * (I_k - Sum_YV_1 - Sum_YV_2); % k모선의 i번째 iteration 값 계산
19 Delta(k,i+1) = atan(imag(V_k) / real(V_k)) * (180/pi);
20 V(k,i+1) = V(k,i); % 원래 PV Bus였으면 PQ Bus로 전환되었다고해도 전압은 유지되어야 함.
21
22 % Recalculation
23 I_k = (P(k,i)-sqrt(-1)*Q(k,i)) / (V(k,i+1)*exp(-sqrt(-1)*Delta(k,i+1)*(pi/180)));
24 V_k = (1/Y(k,k)) * (I_k - Sum_YV_1 - Sum_YV_2);
25 Delta(k,i+1) = atan(imag(V_k) / real(V_k)) * (180/pi);
26
27 end

```

PV Bus의 Q_G 가 limit을 벗어나면 더 이상 PV Bus로서의 역할을 수행할 수 없다고 간주하고 PQ Bus로 스위칭하게 된다. 다만, PQ Bus로 전환한다고 해도 원래의 전압은 유지해야 하므로 온전한 PQ Bus의 알고리즘을 적용할 수 없어 새로운 종류의 모선으로 취급하여 계산하도록 설계하였다.

⑦ Line Flow Calculation

$$\begin{aligned}
 S_{Zn} &= V_n * (V_n^* - V_k^*) * Y_{nk} & S_{Yn} &= V_k * V_k^* * y_{kn} \\
 S_{Zr} &= V_k * (V_k^* - V_n^*) * Y_{kn} & S_{Yr} &= V_n * V_n^* * y_{nk} \\
 S_{nor} &= S_{Zn} + S_{Yn} & S_{rev} &= S_{Zr} + S_{Yr}
 \end{aligned}$$

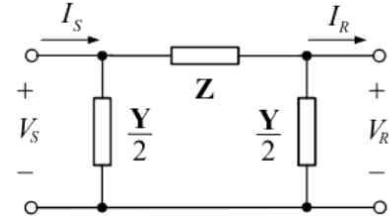


Figure 11. Line Flow Calculation

코드)

```

1 % Line Flow Calculation
2
3 function LineOutputData = Line_Flow_Calc(L_Mat,i,V,Delta,Y)
4
5     L_Mat = repelem(L_Mat,2,1);
6     [rows,~] = size(L_Mat);
7
8     LineOutputData = zeros(rows,6); % Line Flow Data
9
10    LineOutputData(:,1) = L_Mat(:,1);
11
12    for j = 1:rows % Bus to Bus
13        if mod(j,2) == 0
14            LineOutputData(j,2) = L_Mat(j,3);
15            LineOutputData(j,3) = L_Mat(j,2);
16        else
17            LineOutputData(j,2) = L_Mat(j,2);
18            LineOutputData(j,3) = L_Mat(j,3);
19        end
20    end
21
22    V_vector = V(:,i+1) .* exp(Delta(:,i+1)*(sqrt(-1)*(pi/180)));
23
24    for j = 1:rows % P, Q, S
25
26        E = (V_vector(LineOutputData(j,2)) - V_vector(LineOutputData(j,3))); % Vs - Vr
27        I = conj(-1*(LineOutputData(j,2),LineOutputData(j,3))*conj(E));
28        S = V_vector(LineOutputData(j,2))*I + V_vector(LineOutputData(j,2))*conj(V_vector(LineOutputData(j,2))*(L_Mat(j,6)+L_Mat(j,7)*sqrt(-1))/2);
29
30        LineOutputData(j,4) = real(S); % P
31        LineOutputData(j,5) = imag(S); % Q
32        LineOutputData(j,6) = abs(S); % S
33    end
34 end

```

p.u.값을 사용하므로 송전선의 변압기 유무에 상관없이 똑같은 방식으로 Line Flow를 계산할 수 있다.

4) 다양한 전력 계통의 raw data 제작

임의 개수의 모선을 가지는 다양한 전력 시스템의 Power Flow 계산에 대하여 편리성을 위해 MATLAB에서 EXCEL로 만든 전력 계통의 raw data를 읽는 방식을 선택하였다.

```
% import Bus Data File
% bus_file = 'example6.10_bus.xlsx';
% bus_file = 'example6.38_bus.xlsx';
% bus_file = 'ieee5bus_bus.xlsx';
bus_file = 'ieee9bus_bus.xlsx';
% bus_file = 'ieee14bus_bus.xlsx';
% bus_file = 'ieee30bus_bus.xlsx';

% import Line Data File
% line_file = 'example6.10_line.xlsx';
% line_file = 'example6.38_line.xlsx';
% line_file = 'ieee5bus_line.xlsx';
line_file = 'ieee9bus_line.xlsx';
% line_file = 'ieee14bus_line.xlsx';
% line_file = 'ieee30bus_line.xlsx';

% Initialize Values
[SIZE,V,Delta,P_G,Q_G,P_L,Q_L,Q_Gmax,Q_Gmin,P,Q,Bus_Type,Switch_Sig] = Init_Value(bus_file,ITERATION);

% Set Line Data (Combination of TL & TR data)
[L_Mat,L_data] = import_L_Data(line_file,SIZE);
```

Main Code 내에서 EXCEL 파일을 읽어 Bus의 Initial Value를 지정하고 Line data를 설정한다.

EXCEL로 구성한 전력 계통의 raw data는 다음과 같다.

<2모선 - 교재 EX 6.38>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1	0			0	0		0
3	2				0	0	1.8	0.6		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	1	2	0	0.1	0	0	0	0

<5모선(1) - 교재 EX 6.10>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1	0				0	0	
3	2				0	0	8	2.8		
4	3		1.05	0	5.2	0	0.8	0.4	4	-2.8
5	4				0	0	0	0		
6	5				0	0	0	0		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	2	4	0.009	0.1	0	1.72	12	0
3	2	2	5	0.0045	0.05	0	0.88	12	0
4	3	4	5	0.00225	0.025	0	0.44	12	0
5	4	1	5	0.0015	0.02	0	0	6	1
6	5	3	4	0.00075	0.01	0	0	10	1

<5모선(2) - IEEE 5 Bus>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1.06	0				0	0	
3	2		1		0.4	0.3	0.2	0.1	9	-9
4	3				0	0	0.45	0.15		
5	4				0	0	0.4	0.05		
6	5				0	0	0.6	0.1		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	1	2	0.02	0.06	0	0	0.8	0
3	2	1	3	0.08	0.24	0	0.025	0.3	0
4	3	2	3	0.06	0.25	0	0.02	0.2	0
5	4	2	4	0.06	0.18	0	0.02	0.2	0
6	5	2	5	0.04	0.12	0	0.015	0.6	0
7	6	3	4	0.01	0.03	0	0.01	0.1	0
8	7	4	5	0.08	0.24	0	0.025	0.1	0

<9모선 - IEEE 9 Bus>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1.04	0			0	0		
3	2		1.025		1.63	0	0	0	3	-3
4	3		1.025		0.85	0	0	0	3	-3
5	4				0	0	0	0		
6	5				0	0	1.25	0.5		
7	6				0	0	0.9	0.3		
8	7				0	0	0	0		
9	8				0	0	1	0.35		
10	9				0	0	0	0		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	1	4	0	0.0576	0	0	0	1
3	2	4	6	0.017	0.092	0	0.158	0	0
4	3	6	9	0.039	0.17	0	0.358	0	0
5	4	3	9	0	0.0586	0	0	0	1
6	5	9	8	0.0119	0.1008	0	0.209	0	0
7	6	8	7	0.0085	0.072	0	0.149	0	0
8	7	7	2	0	0.0625	0	0	0	1
9	8	7	5	0.032	0.161	0	0.306	0	0
10	9	5	4	0.01	0.085	0	0.176	0	0

<14모선 - IEEE 14 Bus>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1.06	0			0	0		
3	2		1.045		0.4	0.424	0.217	0.127	0.5	-0.4
4	3		1.01		0	0.234	0.942	0.19	0.4	0
5	4				0	0	0.478	0.039		
6	5				0	0	0.076	0.016		
7	6		1.07		0	0.122	0.112	0.075	0.24	-0.06
8	7				0	0	0	0		
9	8		1.09		0	0.174	0	0	0.24	-0.06
10	9				0	0	0.295	0.166		
11	10				0	0	0.09	0.058		
12	11				0	0	0.035	0.018		
13	12				0	0	0.061	0.016		
14	13				0	0	0.135	0.058		
15	14				0	0	0.149	0.05		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	1	2	0.01938	0.05917	0	0.0528	0	0
3	2	1	5	0.05403	0.22304	0	0.0492	0	0
4	3	2	3	0.04699	0.19797	0	0.0438	0	0
5	4	2	4	0.05811	0.17632	0	0.034	0	0
6	5	2	5	0.05695	0.17388	0	0.0346	0	0
7	6	3	4	0.06701	0.17103	0	0.0128	0	0
8	7	4	5	0.01335	0.04211	0	0.0001	0	0
9	8	4	7	0	0.20912	0	0	0	0.978
10	9	4	9	0	0.55618	0	0	0	0.969
11	10	5	6	0	0.25202	0	0	0	0.932
12	11	6	11	0.09498	0.1989	0	0.0001	0	0
13	12	6	12	0.12291	0.25581	0	0.0001	0	0
14	13	6	13	0.06615	0.13027	0	0.0001	0	0
15	14	7	8	0.0001	0.17615	0	0.0001	0	0
16	15	7	9	0.0001	0.11001	0	0.0001	0	0
17	16	9	10	0.03181	0.0845	0	0.0001	0	0
18	17	9	14	0.12711	0.27038	0	0.0001	0	0
19	18	10	11	0.08205	0.19207	0	0.0001	0	0
20	19	12	13	0.22092	0.19988	0	0.0001	0	0
21	20	13	14	0.17093	0.34802	0	0.0001	0	0

	A	B	C	D	E	F	G	H	I
1	Line Num	from	to	R	X	G	B	maxMVA	TAP
2	1	1	4	0	0.0576	0	0	0	1
3	2	4	6	0.017	0.092	0	0.158	0	0
4	3	6	9	0.039	0.17	0	0.358	0	0
5	4	3	9	0	0.0586	0	0	0	1
6	5	9	8	0.0119	0.1008	0	0.209	0	0
7	6	8	7	0.0085	0.072	0	0.149	0	0
8	7	7	2	0	0.0625	0	0	0	1
9	8	7	5	0.032	0.161	0	0.306	0	0
10	9	5	4	0.01	0.085	0	0.176	0	0

<30모선 - IEEE 30 Bus>

Bus data)

	A	B	C	D	E	F	G	H	I	J
1	Bus Num	Bus Type	V	Delta	Pg	Qg	PL	QL	Q_Gmax	Q_Gmin
2	1		1.06	0			0	0		
3	2		1.04313		0.4	0.5	0.217	0.127	9	-9
4	3				0	0	0.024	0.012		
5	4				0	0	0.076	0.016		
6	5		1.011		0	0.3685	0.942	0.19	9	-9
7	6				0	0	0	0		
8	7				0	0	0.228	0.109		
9	8		1.01		0	0.3714	0.3	0.3	9	-9
10	9				0	0	0	0		
11	10				0	0	0.058	0.02		
12	11		1.082		0	0.1617	0	0	9	-9
13	12				0	0	0.112	0.075		
14	13		1.071		0	0.1062	0	0	9	-9
15	14				0	0	0.062	0.016		
16	15				0	0	0.082	0.025		
17	16				0	0	0.035	0.018		
18	17				0	0	0.09	0.058		
19	18				0	0	0.032	0.009		
20	19				0	0	0.095	0.034		
21	20				0	0	0.022	0.007		
22	21				0	0	0.175	0.112		
23	22				0	0	0	0		
24	23				0	0	0.032	0.016		
25	24				0	0	0.087	0.067		
26	25				0	0	0	0		
27	26				0	0	0.035	0.023		
28	27				0	0	0	0		
29	28				0	0	0	0		
30	29				0	0	0.024	0.009		
31	30				0	0	0.106	0.019		

Line data)

	A	B	C	D	E	F	G	H	I
1	Line Num from	to	R	X	G	B	maxMVA	TAP	
2	1	1	2	0.0192	0.0575	0	0.0528	0	0
3	2	1	3	0.0452	0.165	0	0.0408	0	0
4	3	2	4	0.057	0.174	0	0.0368	0	0
5	4	2	5	0.0472	0.198	0	0.0418	0	0
6	5	2	6	0.0581	0.176	0	0.0374	0	0
7	6	3	4	0.0132	0.0379	0	0.0084	0	0
8	7	4	6	0.0119	0.0414	0	0.009	0	0
9	8	5	7	0.046	0.116	0	0.0204	0	0
10	9	6	7	0.0267	0.082	0	0.017	0	0
11	10	6	8	0.012	0.042	0	0.009	0	0
12	11	6	28	0.0169	0.0599	0	0.013	0	0
13	12	8	28	0.0636	0.2	0	0.0428	0	0
14	13	9	10	0.0001	0.11	0	0.0001	0	0
15	14	9	11	0.0001	0.208	0	0.0001	0	0
16	15	10	17	0.0324	0.0845	0	0.0001	0	0
17	16	10	20	0.0936	0.209	0	0.0001	0	0
18	17	10	21	0.0348	0.0749	0	0.0001	0	0
19	18	10	22	0.0727	0.15	0	0.0001	0	0
20	19	12	13	0.0001	0.14	0	0.0001	0	0

22	21	12	15	0.0662	0.13	0	0.0001	0	0
23	22	12	16	0.0945	0.199	0	0.0001	0	0
24	23	14	15	0.221	0.2	0	0.0001	0	0
25	24	15	18	0.107	0.219	0	0.0001	0	0
26	25	15	23	0.1	0.202	0	0.0001	0	0
27	26	16	17	0.0524	0.192	0	0.0001	0	0
28	27	18	19	0.0639	0.129	0	0.0001	0	0
29	28	19	20	0.034	0.068	0	0.0001	0	0
30	29	21	22	0.0116	0.0236	0	0.0001	0	0
31	30	22	24	0.115	0.179	0	0.0001	0	0
32	31	23	24	0.132	0.27	0	0.0001	0	0
33	32	24	25	0.189	0.329	0	0.0001	0	0
34	33	25	26	0.254	0.38	0	0.0001	0	0
35	34	25	27	0.109	0.209	0	0.0001	0	0
36	35	27	29	0.22	0.415	0	0.0001	0	0
37	36	27	30	0.32	0.603	0	0.0001	0	0
38	37	29	30	0.24	0.453	0	0.0001	0	0
39	38	4	12	0	0.256	0	0	0	0.932
40	39	27	28	0	0.396	0	0	0	0.968
41	40	6	10	0	0.556	0	0	0	0.969
42	41	6	9	0	0.208	0	0	0	0.978

```

BusType_Init.m x +
1 % BusType Initialization
2
3 function [raw_B_data] = BusType_Init(raw_B_data,SIZE)
4
5 for i = 1:SIZE
6     if isnan(raw_B_data(i,3)) && isnan(raw_B_data(i,4)) % PQ Bus
7         raw_B_data(i,2) = 2; % Bus Type
8         raw_B_data(i,3) = 1.0; % V
9         raw_B_data(i,4) = 0; % Phase
10        raw_B_data(i,9) = 0; raw_B_data(i,10) = 0; % Q_G limit
11
12        elseif ~(isnan(raw_B_data(i,9)) || isnan(raw_B_data(i,10))) % PV Bus
13            raw_B_data(i,2) = 1; % Bus Type
14            raw_B_data(i,4) = 0; % Phase
15
16        else % Slack Bus
17            raw_B_data(i,2) = 0; % Bus Type
18            raw_B_data(i,5) = 0; % P_G
19            raw_B_data(i,6) = 0; % Q_G
20            raw_B_data(i,9) = 0; raw_B_data(i,10) = 0; % Q_G limit
21        end
22    end
23
24 end

```

모선 종류별 기지값 (Known Value) 특징을 바탕으로 Bus Type을 자동으로 구분하는 코드이며 Init_Value.m 코드 내에서 수행된다.

a) 모선 구분 방법

PQ Bus(2) : 모선의 V 크기와 위상값이 입력되어 있지 않을 때

PV Bus(1) : 모선의 Q_G limit이 입력되어 있을 때

Slack bus(0) : 위의 두 가지 경우에 해당하지 않는 모선

b) 모선별 초기화 값

PQ Bus : PQ Bus는 $V = 1.0$, $\delta = 0$, Q_G 의 max, min = 0으로 설정

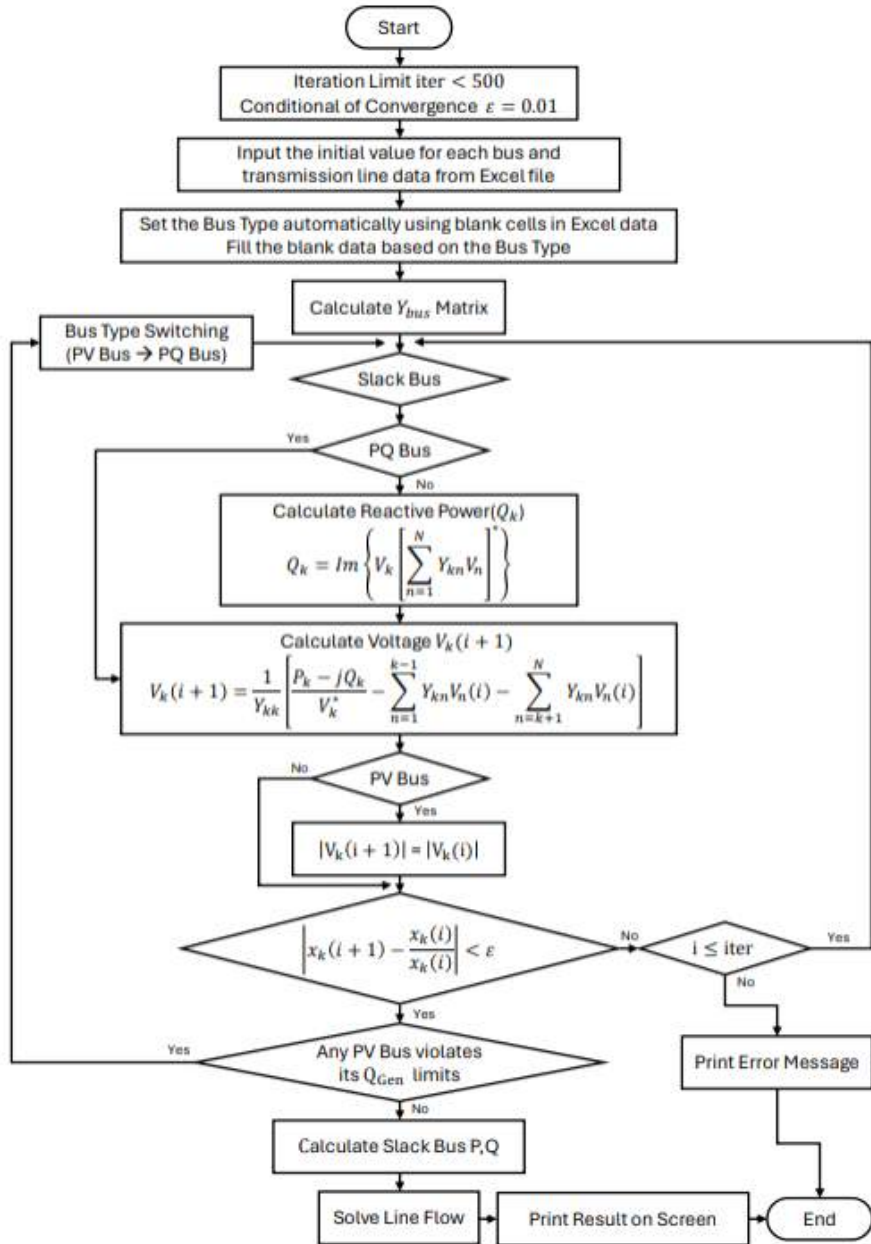
PV Bus : $\delta = 0$ 으로 설정

Slack Bus : P_G , Q_G , Q_G 의 limit을 모두 0으로 설정

위와 같은 과정을 통해 엑셀에 입력되어 있지 않은 모선의 정보를 자동으로 입력할 수 있다.

5) 알고리즘 성능 검증 및 결과 비교

(1) Flow Chart



테스트 환경에서 최대 반복 횟수와 근사 백분율 상대 오차를 지정할 수 있다. Bus Type에 따라 계산이 수행되며 최종 수렴값에서 PV Bus의 Q_G 값이 limit을 벗어나는 경우 해당 Bus를 PQ Bus로 전환하여 처음부터 다시 계산을 수행하도록 코드를 구성하였다.

(2) 코드 실행 결과

① 2-Bus System (교재 예제)

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1	1.0000	0.0000	1.8000	1.0301	0.0000	0.0000
2	0.9149	-11.3469	0.0000	0.0000	1.8000	0.6000
----- TOTAL -----			1.8000	1.0301	1.8000	0.6000

<Line Output Data for the Power System>

Line#	Bus to Bus	P	Q	S
1	1 2	1.8000	1.0301	2.0739
1	2 1	-1.8000	-0.6000	1.8974

② 5-Bus System (교재 예제)

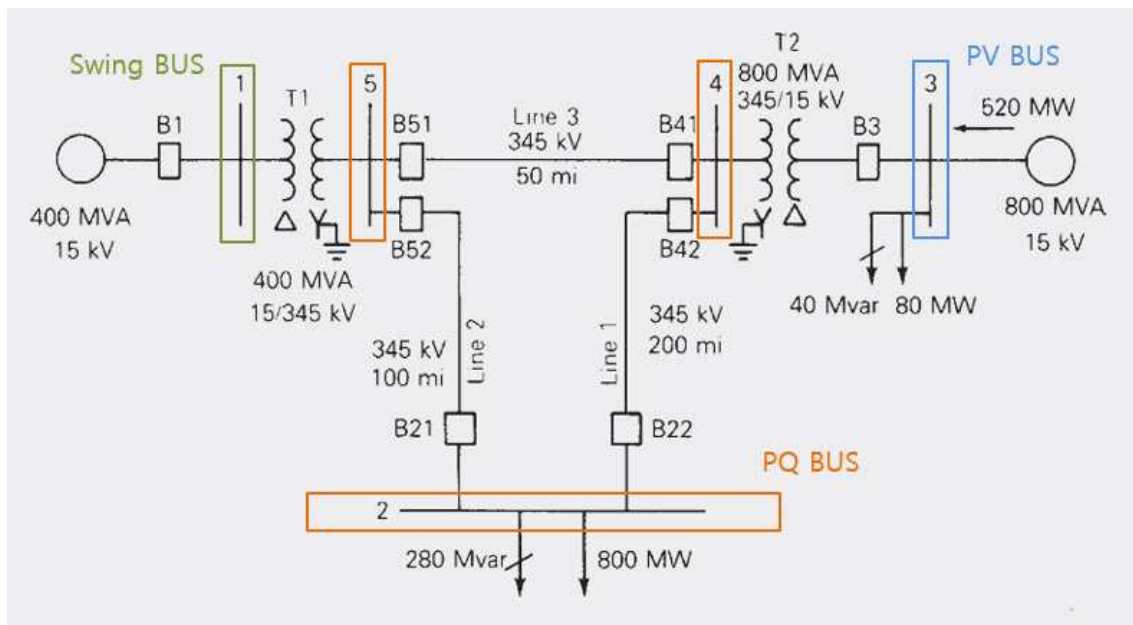


Figure 12. Power System Analysis & Design 6th ed. EX 6.10

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P _G (p.u.)	Q _G (p.u.)	P _L (p.u.)	Q _L (p.u.)
1	1.0000	0.0000	3.9484	1.1428	0.0000	0.0000
2	0.8338	-22.4064	0.0000	0.0000	8.0000	2.8000
3	1.0500	-0.5973	5.2000	3.3748	0.8000	0.4000
4	1.0193	-2.8340	0.0000	0.0000	0.0000	0.0000
5	0.9743	-4.5479	0.0000	0.0000	0.0000	0.0000
----- TOTAL -----			9.1484	4.5176	8.8000	3.2000

<Line Output Data for the Power System>

Line#	Bus to Bus	P	Q	S
1	2 4	-2.9184	-1.3911	3.2330
1	4 2	3.0368	1.2154	3.2710
2	2 5	-5.0816	-1.4089	5.2733
2	5 2	5.2566	2.6302	5.8779
3	4 5	1.3440	1.5035	2.0167
3	5 4	-1.3336	-1.8253	2.2606
4	1 5	3.9484	1.1428	4.1105
4	5 1	-3.9230	-0.8049	4.0048
5	3 4	4.4000	2.9748	5.3113
5	4 3	-4.3808	-2.7189	5.1560

Bus #	Voltage Magnitude (per unit)	Phase Angle (degrees)	Generation		Load	
			PG (per unit)	QG (per unit)	PL (per unit)	QL (per unit)
1	1.000	0.000	3.948	1.144	0.000	0.000
2	0.834	-22.407	0.000	0.000	8.000	2.800
3	1.050	-0.597	5.200	3.376	0.800	0.400
4	1.019	-2.834	0.000	0.000	0.000	0.000
5	0.974	-4.548	0.000	0.000	0.000	0.000
TOTAL			9.148	4.516	8.800	3.200

TABLE 6.6

Bus output data for the power system given in Example 6.9

(Continued)

Line #	Bus to Bus	P	Q	S
1	2 4	-2.920	-1.392	3.232
	4 2	3.036	1.216	3.272
2	2 5	-5.080	-1.408	5.272
	5 2	5.256	2.632	5.876
3	4 5	1.344	1.504	2.016
	5 4	-1.332	-1.824	2.260

TABLE 6.7

Line output data for the power system given in Example 6.9

Tran. #	Bus to Bus	P	Q	S
1	1 5	3.948	1.144	4.112
	5 1	-3.924	-0.804	4.004
2	3 4	4.400	2.976	5.312
	4 3	-4.380	-2.720	5.156

TABLE 6.8

Transformer output data for the power system given in Example 6.9

비교 결과, 교재에서 제공하는 정답과 코드 실행 결과가 일치함을 확인하였다.

③ IEEE 5 Bus System

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1	1.0600	0.0000	1.3122	0.9734	0.0000	0.0000
2	1.0000	-2.0066	0.4000	-0.4974	0.2000	0.1000
3	0.9833	-4.8137	0.0000	0.0000	0.4500	0.1500
4	0.9801	-5.0757	0.0000	0.0000	0.4000	0.0500
5	0.9687	-5.7467	0.0000	0.0000	0.6000	0.1000
----- TOTAL -----			1.7122	0.4760	1.6500	0.4000

<Line Output Data for the Power System>

Line#	Bus to Bus	P	Q	S
1	1 2	0.8780	0.7782	1.1732
1	2 1	-0.8535	-0.7047	1.1068
2	1 3	0.4342	0.1952	0.4760
2	3 1	-0.4177	-0.1717	0.4516
3	2 3	0.1983	0.0138	0.1988
3	3 2	-0.1960	-0.0235	0.1974
4	2 4	0.2978	0.0088	0.2980
4	4 2	-0.2925	-0.0124	0.2927
5	2 5	0.5573	0.0846	0.5637
5	5 2	-0.5445	-0.0609	0.5479
6	3 4	0.1636	0.0452	0.1697
6	4 3	-0.1633	-0.0539	0.1720
7	4 5	0.0558	0.0164	0.0581
7	5 4	-0.0555	-0.0391	0.0679

④ IEEE 9 Bus System

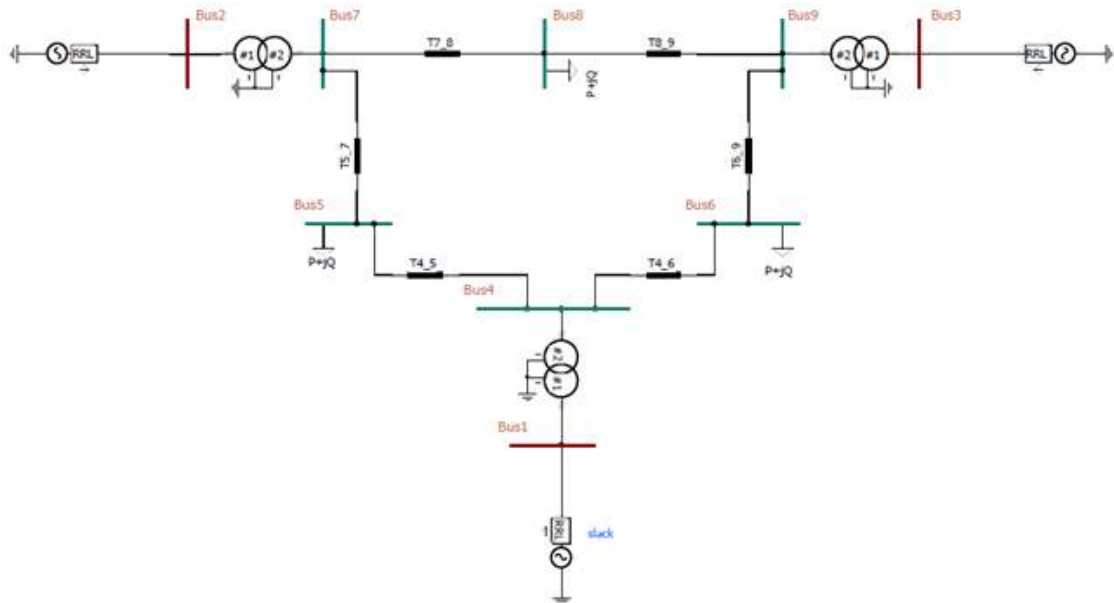


Figure 13. IEEE 9 Bus System - PSCAD

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1	1.0400	0.0000	0.7164	0.2705	0.0000	0.0000
2	1.0250	9.2800	1.6300	0.0665	0.0000	0.0000
3	1.0250	4.6647	0.8500	-0.1086	0.0000	0.0000
4	1.0258	-2.2168	0.0000	0.0000	0.0000	0.0000
5	0.9956	-3.9888	0.0000	0.0000	1.2500	0.5000
6	1.0127	-3.6874	0.0000	0.0000	0.9000	0.3000
7	1.0258	3.7197	0.0000	0.0000	0.0000	0.0000
8	1.0159	0.7275	0.0000	0.0000	1.0000	0.3500
9	1.0324	1.9667	0.0000	0.0000	0.0000	0.0000
----- TOTAL -----			3.1964	0.2284	3.1500	1.1500

<Line Output Data for the Power System>

Line#	Bus to Bus	P	Q	S
1	1 4	0.7164	0.2705	0.7658
1	4 1	-0.7164	-0.2392	0.7553
2	4 6	0.3070	0.0103	0.3072
2	6 4	-0.3054	-0.1654	0.3473
3	6 9	-0.5946	-0.1346	0.6097
3	9 6	0.6082	-0.1807	0.6345
4	3 9	0.8500	-0.1086	0.8569
4	9 3	-0.8500	0.1496	0.8631
5	9 8	0.2418	0.0312	0.2438
5	8 9	-0.2410	-0.2430	0.3422
6	8 7	-0.7590	-0.1070	0.7666
6	7 8	0.7638	-0.0080	0.7638
7	7 2	-1.6300	0.0918	1.6326
7	2 7	1.6300	0.0665	1.6314
8	7 5	0.8662	-0.0838	0.8702
8	5 7	-0.8432	-0.1131	0.8508
9	5 4	-0.4068	-0.3869	0.5614
9	4 5	0.4094	0.2289	0.4690

Table 4 - Source and line power flow comparison of IEEE 9-bus system

Bus	PSS/E		PSCAD	
	P [pu]	Q [pu]	P [pu]	Q [pu]
1	0.716	0.279	0.7152	0.2761
2	1.630	0.049	1.6320	0.0454
3	0.850	-0.114	0.8512	-0.1170
From Bus	To Bus			
4	5	0.433	0.235	0.4322 0.2334
4	6	0.283	0.013	0.2830 0.0115
5	7	0.842	-0.104	0.8430 -0.1041
6	9	0.633	-0.178	0.6340 -0.1810
7	8	0.788	-0.008	0.7892 -0.0089
8	9	0.217	0.023	0.2172 0.0229

Bus		Power at Bus & Line Flow			
From	To	MW[pu]	Mvar[pu]	MVA[pu]	Line Loss[pu]
1	4	0.720	0.270	0.769	0.000
4	1	-0.720	-0.239	0.759	0.000
4	6	0.308	0.010	0.309	0.000
6	4	-0.307	-0.165	0.348	0.002
6	9	-0.593	-0.135	0.609	0.000
9	6	0.607	-0.181	0.633	0.013
3	9	0.849	-0.109	0.856	0.000
9	3	-0.849	0.150	0.862	0.000
9	8	0.242	0.031	0.244	0.000
8	9	-0.241	-0.243	0.342	0.001
8	7	-0.759	-0.107	0.767	0.000
7	8	0.764	-0.008	0.764	0.005
7	2	-1.629	0.092	1.632	0.000
2	7	1.629	0.066	1.631	0.000
7	5	0.865	-0.084	0.869	0.000
5	7	-0.842	-0.113	0.849	0.023
5	4	-0.408	-0.387	0.562	0.000
4	5	0.411	0.229	0.470	0.003

코드 실행 결과가 실제값과 유사하게 출력되었음을 확인하였다.

⑤ IEEE 14 Bus System

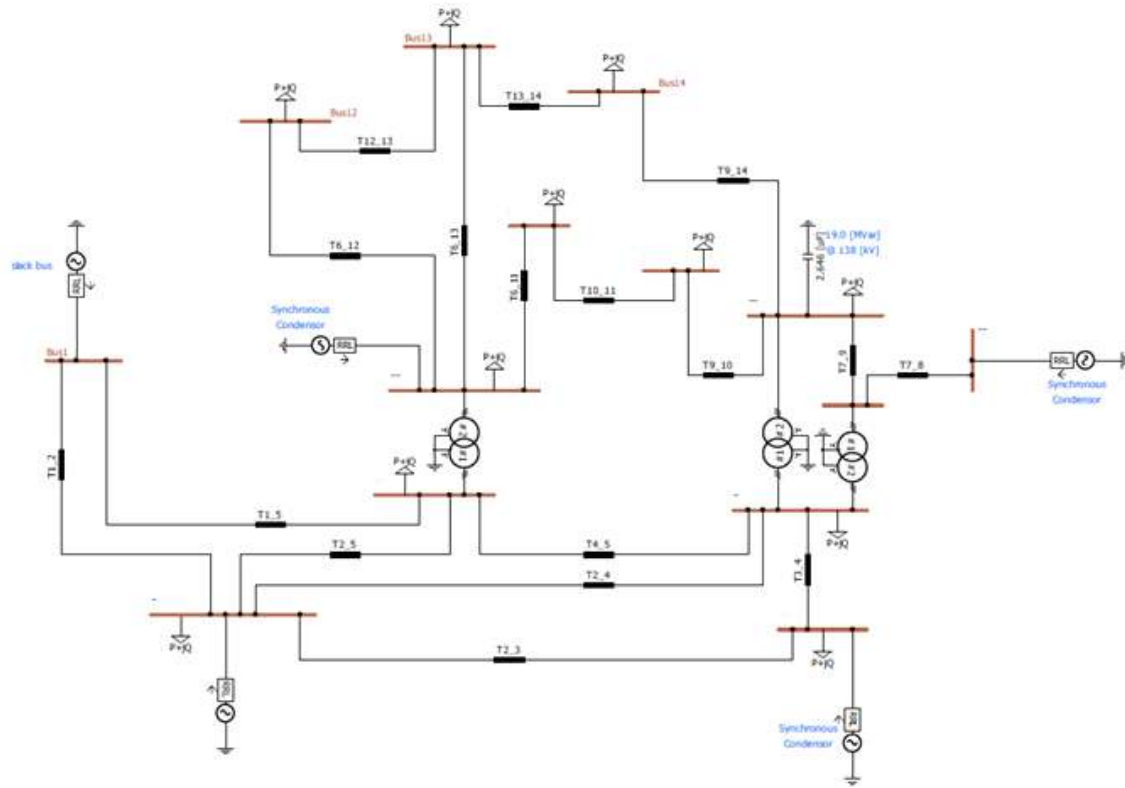


Figure 14. IEEE 14 Bus System - PSCAD

<Bus Output Data for the Power System>

Bus#	Voltage	Magnitude(p.u.)	Phase	Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1		1.0600		0.0000	2.2255	-0.2051	0.0000	0.0000
2		1.0450		-4.7514	0.4000	0.3095	0.2170	0.1270
3		1.0100		-12.3143	0.0000	0.2129	0.9420	0.1900
4		1.0241		-9.8924	0.0000	0.0000	0.4780	0.0390
5		1.0322		-8.4888	0.0000	0.0000	0.0760	0.0160
6		1.0700		-13.6122	0.0000	0.2400	0.1120	0.0750
7		1.0435		-12.7779	0.0000	0.0000	0.0000	0.0000
8		1.0900		-12.7791	0.0000	0.2400	0.0000	0.0000
9		1.0258		-14.2921	0.0000	0.0000	0.2950	0.1660
10		1.0259		-14.4528	0.0000	0.0000	0.0900	0.0580
11		1.0440		-14.1477	0.0000	0.0000	0.0350	0.0180
12		1.0530		-14.4875	0.0000	0.0000	0.0610	0.0160
13		1.0459		-14.5391	0.0000	0.0000	0.1350	0.0580
14		1.0162		-15.4336	0.0000	0.0000	0.1490	0.0500
----- TOTAL -----					2.6255	0.7972	2.5900	0.8130

<Line Output Data for the Power System>

Line#	Bus to	Bus	P	Q	S
1	1	2	1.4989	-0.1876	1.5106
1	2	1	-1.4598	0.2487	1.4808
2	1	5	0.7265	-0.0176	0.7268
2	5	1	-0.7012	0.0685	0.7045
3	2	3	0.7162	0.0372	0.7172
3	3	2	-0.6940	0.0102	0.6940
4	2	4	0.5347	-0.0466	0.5367
4	4	2	-0.5194	0.0565	0.5225
5	2	5	0.3919	-0.0569	0.3960
5	5	2	-0.3838	0.0442	0.3864
6	3	4	-0.2480	0.0127	0.2484
6	4	3	0.2521	-0.0156	0.2526
7	4	5	-0.6129	0.0062	0.6129
7	5	4	0.6176	0.0087	0.6177
8	4	7	0.2573	-0.0887	0.2721
8	7	4	-0.2573	0.1034	0.2773
9	4	9	0.1449	0.0025	0.1449
9	9	4	-0.1449	0.0086	0.1452
10	5	6	0.3913	-0.1375	0.4148
10	6	5	-0.3913	0.1782	0.4300
11	6	11	0.0972	0.0936	0.1349
11	11	6	-0.0957	-0.0905	0.1317
12	6	12	0.0827	0.0319	0.0886
12	12	6	-0.0818	-0.0303	0.0872
13	6	13	0.1909	0.1022	0.2165
13	13	6	-0.1882	-0.0970	0.2117
14	7	8	-0.0000	-0.2753	0.2753
14	8	7	0.0000	0.2874	0.2874
15	7	9	0.2573	0.1718	0.3094
15	9	7	-0.2573	-0.1622	0.3042
16	9	10	0.0300	-0.0131	0.0327
16	10	9	-0.0300	0.0130	0.0327
17	9	14	0.0772	0.0007	0.0772
17	14	9	-0.0764	0.0007	0.0764
18	10	11	-0.0600	-0.0710	0.0930
18	11	10	0.0607	0.0725	0.0946
19	12	13	0.0208	0.0143	0.0253
19	13	12	-0.0207	-0.0143	0.0251
20	13	14	0.0739	0.0533	0.0911
20	14	13	-0.0726	-0.0507	0.0885

Table 4 - Source and line power comparison of IEEE 14-bus system

Bus		PSS/E		PSCAD	
		P [pu]	Q [pu]	P [pu]	Q [pu]
1		2.324	-0.165	2.3230	-0.1548
2		0.400	0.436	0.3995	0.4493
3		0.000	0.251	0.0007	0.2613
6		0.000	0.127	0.0020	0.1498
8		0.000	0.176	-0.0011	0.1896
From Bus	To Bus				
1	2	1.569	-0.204	1.5690	-0.2005
1	5	0.755	0.039	0.7543	0.0450
2	3	0.709	-0.016	0.7096	-0.0164
2	4	0.561	-0.030	0.5606	-0.0209
2	5	0.406	0.012	0.4043	0.0165
3	4	0.237	-0.048	0.2354	-0.0540
4	5	0.612	-0.158	0.6130	-0.1750
6	11	0.074	0.034	0.0747	0.0384
6	12	0.078	0.025	0.0781	0.0253
6	13	0.177	0.072	0.1782	0.0740
7	8	0.000	0.176	0.0011	0.1844
7	9	0.281	0.050	0.2793	-0.0539
9	10	0.052	0.042	0.0511	0.0380
9	14	0.093	0.034	0.0878	0.0217
10	11	0.038	0.016	0.0390	0.0200
12	13	0.016	0.008	0.0166	0.0080
13	14	0.056	0.017	0.0568	0.0188

Bus		Power at Bus & Line Flow			Line Loss[pu]
From	To	MW[pu]	Mvar[pu]	MVA[pu]	
1	2	1.464	-0.149	1.471	
2	1	-1.426	0.263	1.450	-0.037
1	5	0.707	0.010	0.707	
5	1	-0.683	0.089	0.688	-0.024
2	3	0.676	0.065	0.680	
3	2	-0.657	0.018	0.657	-0.020
2	4	0.516	-0.026	0.517	
4	2	-0.502	0.069	0.506	-0.014
2	5	0.379	-0.037	0.381	
5	2	-0.372	0.060	0.377	-0.008
3	4	-0.226	0.007	0.226	
4	3	0.229	0.001	0.229	0.003
4	5	-0.587	-0.002	0.587	
5	4	0.592	0.016	0.592	0.004
4	7	0.245	-0.043	0.249	
7	4	-0.245	0.055	0.251	0.000
4	9	0.138	0.039	0.143	
9	4	-0.138	-0.028	0.140	0.000
5	6	0.387	-0.136	0.410	
6	5	-0.387	0.176	0.425	0.000
6	11	0.109	0.133	0.172	
11	6	-0.106	-0.128	0.167	-0.002
6	12	0.085	0.057	0.093	
12	6	-0.084	-0.035	0.091	-0.001
6	13	0.198	0.123	0.233	
13	6	-0.194	-0.117	0.227	-0.003
7	8	0.000	-0.326	0.326	
8	7	0.000	0.344	0.344	0.000
7	9	0.246	0.271	0.366	
9	7	-0.246	-0.257	0.355	0.000
9	10	0.020	-0.049	0.053	
10	9	-0.020	0.049	0.053	0.000
9	14	0.069	-0.022	0.072	
14	9	-0.068	0.024	0.072	-0.001
10	11	-0.070	-0.107	0.128	
11	10	0.071	0.110	0.131	0.001
12	13	0.023	0.019	0.030	
13	12	-0.023	-0.019	0.030	0.000
13	14	0.083	0.078	0.114	
14	13	-0.081	-0.074	0.109	-0.002

IEEE 14 Bus System의 경우 6, 8번 PV Bus의 Q_G 수렴값이 Q_{Gmax} 를 벗어나 PQ Bus로 전환되어 다시 계산되었다. 계산 결과가 실제값과 다소 차이가 있는데, 이러한 문제의 원인은 Bus Switching 과정의 문제가 아닌 Transformer Tap 값 미반영 때문으로 추정된다.

⑥ IEEE 30 Bus System

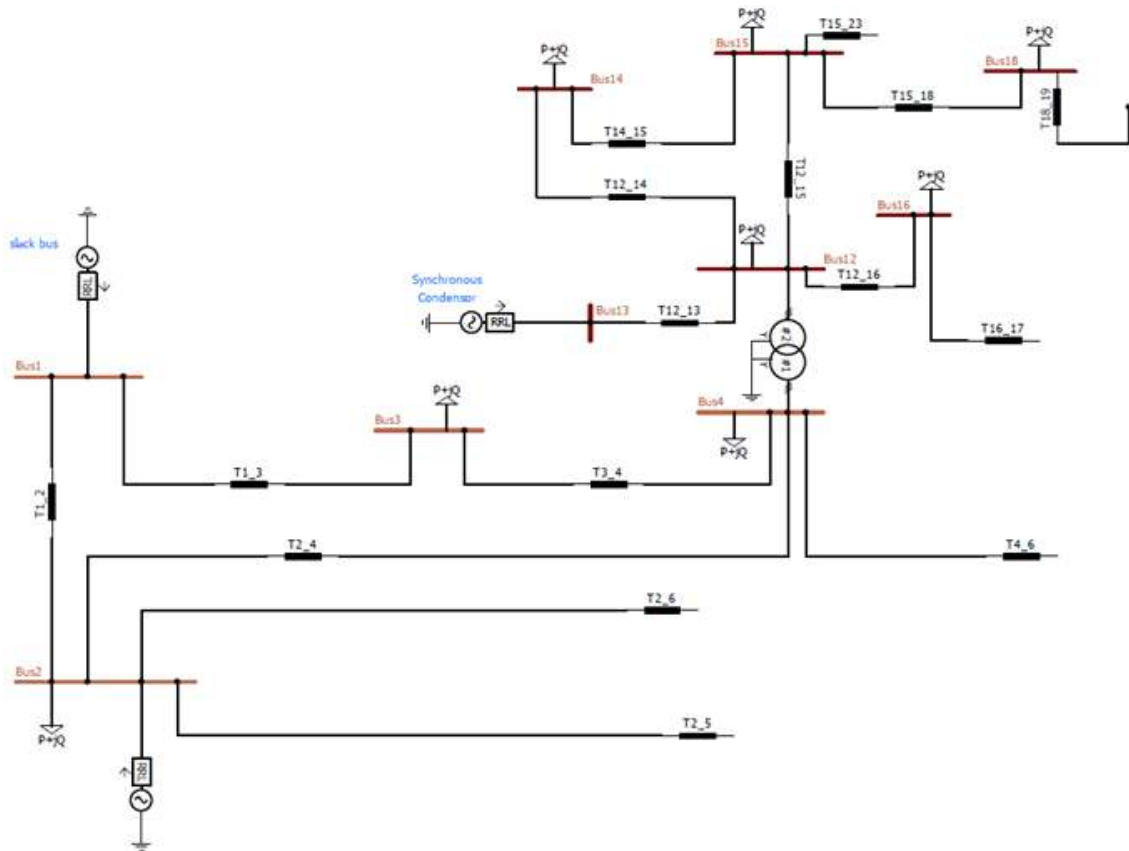


Figure 15. IEEE 30 Bus System - PSCAD

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1	1.0600	0.0000	2.6110	-0.1942	0.0000	0.0000
2	1.0431	-5.3461	0.4000	0.4573	0.2170	0.1270
3	1.0249	-7.5847	0.0000	0.0000	0.0240	0.0120
4	1.0170	-9.3468	0.0000	0.0000	0.0760	0.0160
5	1.0110	-14.1515	0.0000	0.3730	0.9420	0.1900
6	1.0113	-11.0591	0.0000	0.0000	0.0000	0.0000
7	1.0034	-12.8549	0.0000	0.0000	0.2280	0.1090
8	1.0100	-11.7906	0.0000	0.3444	0.3000	0.3000
9	1.0239	-14.2797	0.0000	0.0000	0.0000	0.0000
10	1.0012	-15.9983	0.0000	0.0000	0.0580	0.0200
11	1.0820	-14.2812	0.0000	0.3019	0.0000	0.0000
12	1.0262	-15.5487	0.0000	0.0000	0.1120	0.0750
13	1.0710	-15.5505	0.0000	0.3425	0.0000	0.0000
14	1.0088	-16.4634	0.0000	0.0000	0.0620	0.0160
15	1.0023	-16.4791	0.0000	0.0000	0.0820	0.0250

16		1.0078		-15.9944		0.0000		0.0000		0.0350		0.0180
17		0.9980		-16.2403		0.0000		0.0000		0.0900		0.0580
18		0.9893		-17.0619		0.0000		0.0000		0.0320		0.0090
19		0.9849		-17.2005		0.0000		0.0000		0.0950		0.0340
20		0.9881		-16.9608		0.0000		0.0000		0.0220		0.0070
21		0.9878		-16.4660		0.0000		0.0000		0.1750		0.1120
22		0.9883		-16.4452		0.0000		0.0000		0.0000		0.0000
23		0.9867		-16.7615		0.0000		0.0000		0.0320		0.0160
24		0.9746		-16.7653		0.0000		0.0000		0.0870		0.0670
25		0.9749		-16.3948		0.0000		0.0000		0.0000		0.0000
26		0.9565		-16.8538		0.0000		0.0000		0.0350		0.0230
27		0.9840		-15.8768		0.0000		0.0000		0.0000		0.0000
28		1.0069		-11.6746		0.0000		0.0000		0.0000		0.0000
29		0.9633		-17.2100		0.0000		0.0000		0.0240		0.0090
30		0.9514		-18.1693		0.0000		0.0000		0.1060		0.0190
----- TOTAL -----						3.0110		1.6250		2.8340		1.2620

<Line Output Data for the Power System>

Line#		Bus to Bus			P		Q		S
1		1	2		1.7305		-0.2129		1.7436
1		2	1		-1.6788		0.3094		1.7070
2		1	3		0.8805		0.0187		0.8807
2		3	1		-0.8492		0.0510		0.8508
3		2	4		0.4349		0.0092		0.4350
3		4	2		-0.4250		-0.0179		0.4254
4		2	5		0.8238		0.0129		0.8239
4		5	2		-0.7943		0.0667		0.7971
5		2	6		0.6030		-0.0013		0.6030
5		6	2		-0.5836		0.0207		0.5839
6		3	4		0.8252		-0.0630		0.8276
6		4	3		-0.8166		0.0790		0.8205
7		4	6		0.7252		-0.0641		0.7280
7		6	4		-0.7191		0.0761		0.7231
8		5	7		-0.1477		0.1162		0.1879
8		7	5		0.1494		-0.1326		0.1998
9		6	7		0.3812		-0.0292		0.3823
9		7	6		-0.3774		0.0236		0.3781
10		6	8		0.2962		-0.0548		0.3012
10		8	6		-0.2951		0.0493		0.2992
11		6	28		0.1889		0.0153		0.1895
11		28	6		-0.1883		-0.0265		0.1901
12		8	28		-0.0049		-0.0049		0.0069
12		28	8		0.0049		-0.0386		0.0389
13		9	10		0.2797		0.2160		0.3534
13		10	9		-0.2797		-0.2030		0.3456
14		9	11		0.0000		-0.2858		0.2858
14		11	9		0.0000		0.3019		0.3019
15		10	17		0.0562		0.0163		0.0585
15		17	10		-0.0561		-0.0161		0.0584

16		10	20		0.0898		0.0228		0.0926
16		20	10		-0.0890		-0.0211		0.0914
17		10	21		0.1570		0.1058		0.1893
17		21	10		-0.1557		-0.1032		0.1868
18		10	22		0.0755		0.0497		0.0904
18		22	10		-0.0750		-0.0485		0.0893
19		12	13		0.0000		-0.3283		0.3283
19		13	12		0.0000		0.3425		0.3425
20		12	14		0.0798		0.0318		0.0859
20		14	12		-0.0790		-0.0301		0.0845
21		12	15		0.1787		0.0986		0.2041
21		15	12		-0.1761		-0.0936		0.1994
22		12	16		0.0698		0.0619		0.0933
22		16	12		-0.0691		-0.0604		0.0917
23		14	15		0.0170		0.0141		0.0221
23		15	14		-0.0169		-0.0141		0.0220
24		15	18		0.0608		0.0300		0.0678
24		18	15		-0.0603		-0.0291		0.0670
25		15	23		0.0502		0.0526		0.0727
25		23	15		-0.0497		-0.0517		0.0717
26		16	17		0.0341		0.0424		0.0543
26		17	16		-0.0339		-0.0419		0.0539
27		18	19		0.0283		0.0201		0.0347
27		19	18		-0.0282		-0.0201		0.0346
28		19	20		-0.0668		-0.0139		0.0682
28		20	19		0.0670		0.0141		0.0684
29		21	22		-0.0193		-0.0088		0.0212
29		22	21		0.0193		0.0087		0.0212
30		22	24		0.0557		0.0398		0.0684
30		24	22		-0.0551		-0.0390		0.0675
31		23	24		0.0177		0.0357		0.0398
31		24	23		-0.0175		-0.0353		0.0394
32		24	25		-0.0144		0.0074		0.0162
32		25	24		0.0145		-0.0074		0.0162
33		25	26		0.0355		0.0236		0.0426
33		26	25		-0.0350		-0.0230		0.0419
34		25	27		-0.0500		-0.0163		0.0525
34		27	25		0.0503		0.0168		0.0530
35		27	29		0.0620		0.0167		0.0642
35		29	27		-0.0611		-0.0150		0.0629
36		27	30		0.0710		0.0168		0.0730
36		30	27		-0.0693		-0.0136		0.0706
37		29	30		0.0371		0.0060		0.0376
37		30	29		-0.0367		-0.0054		0.0371
38		4	12		0.4404		-0.0129		0.4406
38		12	4		-0.4404		0.0610		0.4446
39		27	28		-0.1833		-0.0503		0.1901
39		28	27		0.1833		0.0651		0.1945
40		6	10		0.1568		0.0253		0.1588
40		10	6		-0.1568		-0.0116		0.1572
41		6	9		0.2797		-0.0534		0.2848
41		9	6		-0.2797		0.0699		0.2883

Table 4 - Source and line power comparison of IEEE 30-bus system

Bus		PSS/E		PSCAD	
		P [pu]	Q [pu]	P [pu]	Q [pu]
1		2.609	-0.168	2.6070	-0.1530
2		0.400	0.500	0.3992	0.5167
5		0.000	0.369	0.0025	0.3868
8		0.000	0.371	-0.0000	0.4047
11		0.000	0.162	0.0004	0.1662
13		0.000	0.106	0.0009	0.1111
From Bus	To Bus				
1	2	1.732	-0.2130	1.7320	-0.2098
1	3	0.846	-0.0240	0.8442	-0.0128
2	4	0.426	0.0470	0.4253	0.0589
2	5	0.824	0.0180	0.8239	0.0142
2	6	0.603	0.0050	0.6026	0.0169
3	4	0.813	-0.0360	0.8116	-0.0408
4	6	0.715	-0.1760	0.7128	-0.1745
5	7	0.148	-0.1330	0.1469	-0.1465
6	7	0.377	-0.0300	0.3749	-0.0375
6	8	0.296	-0.0810	0.2947	-0.0990
6	28	0.186	0.0110	0.1861	0.0112
8	28	0.005	-0.0040	0.0052	-0.0045
9	10	0.277	0.0590	0.2776	0.0567
9	11	0.000	0.1620	0.0004	0.1662
10	17	0.053	0.0440	0.0531	0.0428
10	20	0.090	0.0370	0.0900	0.0362
10	21	0.157	0.0980	0.1574	0.0980
10	22	0.076	0.0450	0.0754	0.0450
12	13	0.000	0.1060	0.0009	0.1111
12	14	0.078	0.0220	0.0779	0.0227
12	15	0.177	0.0640	0.1771	0.0648
12	16	0.072	0.0340	0.0721	0.0335
14	15	0.016	0.0060	0.0159	0.0067
15	18	0.060	0.0160	0.0590	0.0157
15	23	0.050	0.0290	0.0503	0.0296
16	17	0.037	0.0140	0.0370	0.0154
18	19	0.028	0.0060	0.0279	0.0067
19	20	0.067	0.0280	0.0671	0.0273
21	22	0.018	0.0140	0.0176	0.0138
22	24	0.057	0.0310	0.0573	0.0306
23	24	0.018	0.0120	0.0183	0.0136
24	25	0.012	-0.0200	0.0116	-0.0214
25	26	0.035	0.0230	0.0350	0.0230
25	27	0.048	0.0040	0.0473	0.0025
27	29	0.061	0.0150	0.0610	0.0150
27	30	0.071	0.0170	0.0709	0.0175
29	30	0.037	0.0050	0.0367	0.0054

IEEE 30 Bus System의 경우 Bus Type Switching이 발생하지 않았으며, 코드 실행 결과가 실제값과 유사하게 출력되었음을 확인하였다.

Power Flow 계산에 대한 코드와 Raw Data는 다음 GitHub 링크에서 확인할 수 있다.

https://github.com/hakmin1015/AdvancedPowerFlow_CapstoneProject/tree/main/PowerFlow_Jacobi

⑦ MATLAB 콘솔 창 출력 결과

<2024-1학기 전기공학전공 자기설계학점 : Advanced Power Flow 캡스톤 프로젝트>
[융합전자공학과 201910906 이학민]

ieee5bus_bus.xlsx 파일을 정상적으로 읽었습니다.
ieee5bus_line.xlsx 파일을 정상적으로 읽었습니다.

Iteration Limit : 1000

Threshold Value of Approximate Relative Error[%] : 0.0000001

[Bus Data]

Bus Num	Bus Type	V	Delta	P_G	Q_G	P_L	Q_L	Q_Gmax	Q_Gmin
1.0000	0	1.0600	0	0	0	0	0	0	0
2.0000	1.0000	1.0000	0	0.4000	0.3000	0.2000	0.1000	9.0000	-9.0000
3.0000	2.0000	1.0000	0	0	0	0.4500	0.1500	0	0
4.0000	2.0000	1.0000	0	0	0	0.4000	0.0500	0	0
5.0000	2.0000	1.0000	0	0	0	0.6000	0.1000	0	0

[Transmission Line Data]

Line Num	from	to	Rpu	Xpu	Gpu	Bpu	maxMVA	TAP
1.0000	1.0000	2.0000	0.0200	0.0600	0	0	0.8000	0
2.0000	1.0000	3.0000	0.0800	0.2400	0	0.0250	0.3000	0
3.0000	2.0000	3.0000	0.0600	0.2500	0	0.0200	0.2000	0
4.0000	2.0000	4.0000	0.0600	0.1800	0	0.0200	0.2000	0
5.0000	2.0000	5.0000	0.0400	0.1200	0	0.0150	0.6000	0
6.0000	3.0000	4.0000	0.0100	0.0300	0	0.0100	0.1000	0
7.0000	4.0000	5.0000	0.0800	0.2400	0	0.0250	0.1000	0

[Y Bus Matrix]

6.2500 -18.7375i	-5.0000 +15.0000i	-1.2500 + 3.7500i	0.0000 + 0.0000i	0.0000 + 0.0000i
-5.0000 +15.0000i	10.0744 -31.2546i	-0.9077 + 3.7821i	-1.6667 + 5.0000i	-2.5000 + 7.5000i
-1.2500 + 3.7500i	-0.9077 + 3.7821i	12.1577 -37.5046i	-10.0000 +30.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	-1.6667 + 5.0000i	-10.0000 +30.0000i	12.9167 -38.7225i	-1.2500 + 3.7500i
0.0000 + 0.0000i	-2.5000 + 7.5000i	0.0000 + 0.0000i	-1.2500 + 3.7500i	3.7500 -11.2300i

<Final Approximate Percent Relative Error>

```
err_V(Bus1) = 0.0000000000[%]
err_V(Bus2) = 0.0000000000[%]
err_V(Bus3) = 0.0000000931[%]
err_V(Bus4) = 0.0000000904[%]
err_V(Bus5) = 0.0000000446[%]
```

<Bus Output Data for the Power System>

Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
1	1.0600	0.0000	1.3122	0.9734	0.0000	0.0000
2	1.0000	-2.0066	0.4000	-0.4974	0.2000	0.1000
3	0.9833	-4.8137	0.0000	0.0000	0.4500	0.1500
4	0.9801	-5.0757	0.0000	0.0000	0.4000	0.0500
5	0.9687	-5.7467	0.0000	0.0000	0.6000	0.1000
----- TOTAL -----			1.7122	0.4760	1.6500	0.4000

<Line Output Data for the Power System>

Line#	Bus to Bus	P	Q	S
1	1 2	0.8780	0.7782	1.1732
1	2 1	-0.8535	-0.7047	1.1068
2	1 3	0.4342	0.1952	0.4760
2	3 1	-0.4177	-0.1717	0.4516
3	2 3	0.1983	0.0138	0.1988
3	3 2	-0.1960	-0.0235	0.1974
4	2 4	0.2978	0.0088	0.2980
4	4 2	-0.2925	-0.0124	0.2927
5	2 5	0.5573	0.0846	0.5637
5	5 2	-0.5445	-0.0609	0.5479
6	3 4	0.1636	0.0452	0.1697
6	4 3	-0.1633	-0.0539	0.1720
7	4 5	0.0558	0.0164	0.0581
7	5 4	-0.0555	-0.0391	0.0679

x모선의 n번째 iteration 결과(x,n) / (0,0)을 눌러 종료 : 4,30

<4모선의 30번째 iteration 결과>

Voltage Magnitude(p.u.)	Phase Angle(Deg)	P(p.u.)	Q(p.u.)
0.9803	-4.9445	-0.4000	-0.0500

MATLAB 프로그램 내에서 코드를 실행하면 다음과 같은 출력을 볼 수 있다.

- ㉑ 어떤 파일을 읽었는지 알려주는 문구
- ㉒ 최대 반복 횟수 및 반복 중지를 위한 근사 상대 백분율 오차 기준값
- ㉓ Bus Type 및 변수 초기화 이후의 Bus Data
- ㉔ Excel로부터 불러온 Transmission Line Data
- ㉕ 계산된 Y Bus Matrix
- ㉖ 최종 근사 상대 백분율 오차값
- ㉗ Bus Output Data
- ㉘ Line Output Data
- ㉙ 임의 모선의 n번째 iteration 결과

특히 Y Bus Matrix, Bus Output Data, Line Output Data는 하나의 Excel 파일에 따로 저장할 수 있도록 하여 계산 결과의 효과적인 표출 기능을 구현하였다.

⑧ Export Result to Excel File

코드)

```

1  % Export Ybus, Bus/Line Output data to Excel file
2
3  function export_Result(Ybus,BusOutputData,LineOutputData)
4
5      result_file = 'Power_Flow_Result.xlsx';
6      if exist(result_file, 'file')
7          delete(result_file);
8      end
9
10     % Y Bus Matrix
11     writematrix(Ybus,'Power_Flow_Result.xlsx','sheet','Y Bus Matrix');
12
13     % Bus Output Data
14     headers = {'Bus#','Voltage Magnitude(p.u.)','Phase Angle(Deg)','P_G(p.u.)','Q_G(p.u.)','P_L(p.u.)','Q_L(p.u.)'};
15     BusData = [headers; num2cell(BusOutputData)];
16     writecell(BusData,'Power_Flow_Result.xlsx','sheet','Bus Output Data');
17
18     % Line Output Data
19     headers = {'Line#','from','to','P','Q','S'};
20     LineData = [headers; num2cell(LineOutputData)];
21     writecell(LineData,'Power_Flow_Result.xlsx','sheet','Line Output Data');
22
23 end

```

Excel 화면)

	A	B	C	D	E	F	G	H	I
1	0-17.3611i	0+0i	0+0i	0+17.3611i	0+0i	0+0i	0+0i	0+0i	0+0i
2	0+0i	0-16i	0+0i	0+0i	0+0i	0+0i	0+16i	0+0i	0+0i
3	0+0i	0+0i	0-17.0648i	0+0i	0+0i	0+0i	0+0i	0+0i	0+17.0648i
4	0+17.3611i	0+0i	0+0i	3.30738-39.3089i	-1.36519+11.6041i	-1.94219+10.5107i	0+0i	0+0i	0+0i
5	0+0i	0+0i	0+0i	-1.36519+11.6041i	2.55279-17.3382i	0+0i	-1.1876+5.97513i	0+0i	0+0i
6	0+0i	0+0i	0+0i	-1.94219+10.5107i	0+0i	3.2242-15.8409i	0+0i	0+0i	-1.28201+5.58824i
7	0+0i	0+16i	0+0i	0+0i	-1.1876+5.97513i	0+0i	2.80473-35.4456i	-1.61712+13.698i	0+0i
8	0+0i	0+0i	0+0i	0+0i	0+0i	0+0i	-1.61712+13.698i	2.77221-23.3032i	-1.15509+9.78427i
9	0+0i	0+0i	0+17.0648i	0+0i	0+0i	-1.28201+5.58824i	0+0i	-1.15509+9.78427i	2.4371-32.1539i

Y Bus Matrix

	A	B	C	D	E	F	G
1	Bus#	Voltage Magnitude(p.u.)	Phase Angle(Deg)	P_G(p.u.)	Q_G(p.u.)	P_L(p.u.)	Q_L(p.u.)
2	1	1.04	0	0.71641043	0.270459208	0	0
3	2	1.025	9.280003346	1.63	0.066536586	0	0
4	3	1.025	4.664749223	0.85	-0.108597101	0	0
5	4	1.025788395	-2.216788471	0	0	0	0
6	5	0.995630861	-3.988806446	0	0	1.25	0.5
7	6	1.012654326	-3.687397357	0	0	0.9	0.3
8	7	1.025769373	3.719699124	0	0	0	0
9	8	1.015882584	0.727533959	0	0	1	0.35
10	9	1.03235295	1.966714065	0	0	0	0

Bus Output Data

	A	B	C	D	E	F
1	Line#	from	to	P	Q	S
2	1	1	4	0.716410433	0.270459208	0.765762425
3	1	4	1	-0.716410433	-0.239231227	0.755298278
4	2	4	6	0.307036797	0.010300044	0.307209514
5	2	6	4	-0.305372727	-0.165433629	0.347305036
6	3	6	9	-0.594627289	-0.134566366	0.609663612
7	3	9	6	0.608165774	-0.180748357	0.634456915
8	4	3	9	0.849999969	-0.108597101	0.856909142
9	4	9	3	-0.849999969	0.149553281	0.863056273
10	5	9	8	0.24183416	0.03119508	0.243837843
11	5	8	9	-0.240954194	-0.242958221	0.342180684
12	6	8	7	-0.759045848	-0.107041771	0.766556287
13	6	7	8	0.763798685	-0.007973315	0.763840301
14	7	7	2	-1.629999971	0.0917815	1.632581927
15	7	2	7	1.629999971	0.066536586	1.631357417
16	8	7	5	0.866201246	-0.083808181	0.870246178
17	8	5	7	-0.843201538	-0.113127531	0.850756529
18	9	5	4	-0.406798479	-0.386872464	0.561386948
19	9	4	5	0.409373622	0.228931186	0.469037579

Line Output Data

III. Conclusion

1) 프로젝트의 한계 및 극복방안

학기 내에 한정된 시간에서 프로젝트를 수행하며 효율적인 알고리즘 구성에 대한 깊은 고찰을 하지 못하였다. 그에 따라 알고리즘의 최적화가 되어 있지 않아 코드의 가독성이 나빠지고 시간 복잡도(Time Complexity) 및 공간 복잡도(Space Complexity)가 증가하였다.

실제 산업의 경우 대규모 계통에서의 실시간으로 빠르게 정보 처리가 이루어져야 하므로 알고리즘의 최적화 및 해의 수렴성이 매우 중요하다. 따라서 수렴 속도가 느린 Jacobi Method 보다는 Newton-Raphson Method가 주로 사용되고 있으며 보조적으로 Gauss-Seidel Method가 활용된다. 따라서 추후 코드의 효율성을 개선할 때 새로운 수치해석 기법을 적용할 것을 고려할 수 있다.

프로젝트에서 사용한 예제 중 14 Bus, 30 Bus와 같이 모선의 수가 많은 계통의 경우 Transformer Tap이 적용된다. 본 프로젝트에서는 Transformer Tap을 고려하지 못해 코드 실행 결과와 실제값의 차이가 다소 발생하였다. 이후 개발 과정에서 Transformer Tap을 고려하여 전력 조류를 계산할 수 있도록 코드를 추가해야 할 것이다.

2) 프로젝트 수행 소감 및 기대효과

본 프로젝트의 목표는 전기공학과에서 학습한 전공 내용을 최종적으로 취합, 정리하는 의미에서 학부 수준 이상의 난이도를 지닌 프로젝트형 문제(알고리즘 구현)를 풀고 MATLAB을 활용하여 효과적인 프로그램을 제작 및 발표하는 것이다.

프로젝트를 수행하면서 수치해석과 전력공학에 대한 이해도가 크게 향상되었고 복잡한 알고리즘에 대하여 설계 의도를 명확히 기술하고 상대방이 이해할 수 있게 설명하는 효과적인 방법에 대해 익힐 수 있었다.

끝으로, 팀별 세미나와 통합 세미나를 통해 협업 능력이 강화되었으며 동일 전공자 간의 친목을 다질 수 있는 더할 나위 없이 좋은 기회였다.

3) 향후 계획

- ① Jacobi Method 알고리즘 최적화 및 코드 가독성 개선
- ② 전력 계통 고장 계산을 수행할 수 있는 코드 추가
- ③ Gauss-Seidel 또는 Newton-Raphson 기법 추가 개발
- ④ <대한전기학회 전력기술부문회 추계학술대회 - 전력조류계산 관련 SW경진대회> 출전
(https://www.kiee.or.kr/board/?_0000_method=view&ncode=a002&num=2519&page=1)
- ⑤ 상명대학교 전기공학과 졸업발표회(ECC)에서 자기설계학점 내용에 대한 발표 예정