Washington State University

EE 521 Analysis of Power System

Assignment #1

Power Flow Project

Name: Chufeng Sun

ID: 11701606

Professor: Dr.Anjan Bose

10/09/2021

1. Assignment Overview

In this assignment, we analyzed the power flow of 14 buses' power system. As shown in the figure below, the system has a total of three synchronous condensers and two generators.

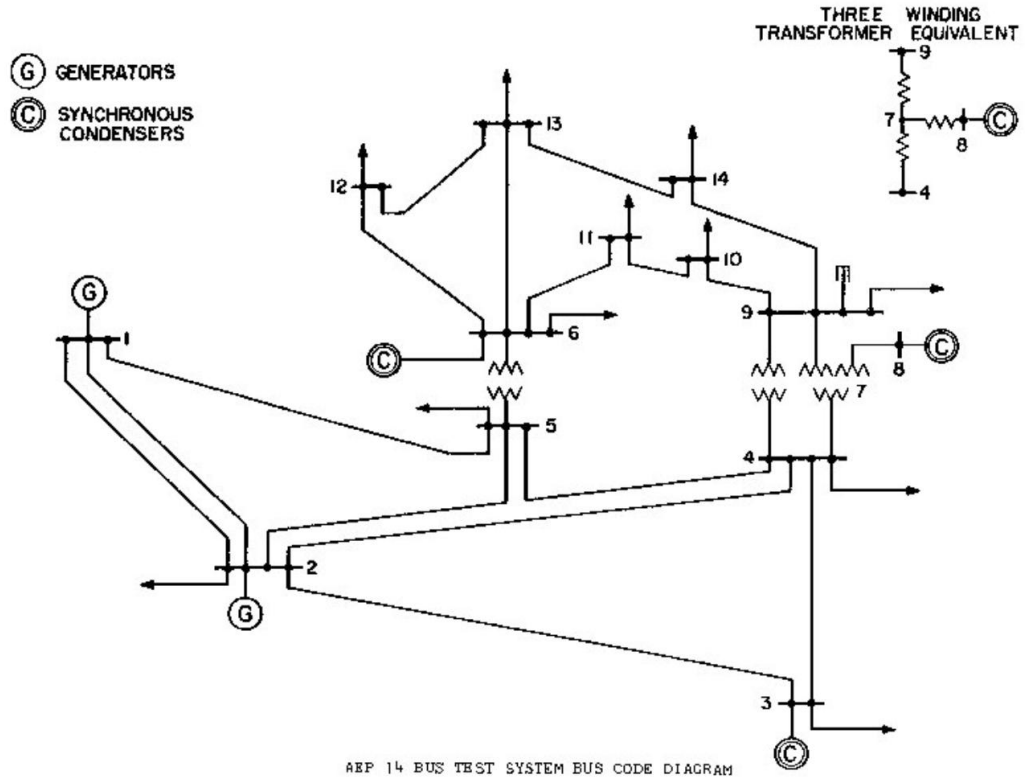

Fig 1. The IEEE 14 Buses System

According to this system, we use Matlab for power flow analysis. Specific analysis steps are as follows:

Method 1: (Normal method)

1. Perform 14buses data import, including bus data and branch data.

2. Y matrix was obtained according to Branch data.

3. Check whether the Q value of PQ bus is within limits.

4. Jacobian matrix is obtained according to bus data.

5. Compare the value of mismatches and determine the times of iterations.

6. Get the final values of V and Vangle.

Method 2: (Factorization, Backward-Forward substitution method)

1. Perform 14buses data import, including bus data and branch data.

2. Y matrix was obtained according to Branch data.

3. Check whether the Q value of PQ bus is within limits.

4. Inverse Jacobian matrix is calculated by factorization.

5.Compare the value of mismatches and determine the times of iterations.

6.Get the final values of V and Vangle.

Method 3: (Fast Decoupled method)

1. Perform 14buses data import, including bus data and branch data.

2. Y matrix was obtained according to Branch data.

3. Check whether the Q value of PQ bus is within limits.

4. B matrix is obtained by Y bus.

5.Compare the value of mismatches and determine the times of iterations.

6.Get the final values of V and Vangle.

2. Power Flow Equations

In this assignment, some nonlinear methods were used to solve the problem of power flow. The specific formula is as follows:

$$S_i = V_i * I_i \qquad (1)$$

For Eq (1), we also can describe as:

$$S = P_i + JQ_i \qquad (2)$$

Also, P and Q can be separately described as:

$$P_i = \sum_{i=1}^{n} V_i V_j Y_{ij} \cos(\delta_i + \delta_j - \theta_{ij}) \qquad (3)$$

$$Q_i = \sum_{i=1}^{n} V_i V_j Y_{ij} \sin(\delta_i + \delta_j - \theta_{ij}) \qquad (4)$$

According Y = G + jB, we can rewrite eq. (3) (4) as:

$$P_i = V_i^2 G_{ii} + \sum_{\substack{i=1 \\ j \neq i}}^{n} V_i V_j Y_{ij} \cos(\delta_i + \delta_j - \theta_{ij}) \qquad (5)$$

$$G_i = -V_i^2 B_{ii} + \sum_{\substack{i=1 \\ j \neq i}}^{n} V_i V_j Y_{ij} \sin(\delta_i + \delta_j - \theta_{ij}) \qquad (6)$$

Then, we use Newton-Raphson method to calculate power flow, the specific formula is as follows:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & V * \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \delta} & V * \frac{\partial Q}{\partial V} \end{bmatrix} * \begin{bmatrix} \Delta \delta \\ \frac{\Delta V}{V} \end{bmatrix} \qquad (7)$$

For Eq. (7), $J_{11} = \frac{\partial P}{\partial \delta}$, where

$$\frac{\partial P_i}{\partial \delta_j} = V_i V_j Y_{ij} \sin(\delta_i + \delta_j - \theta_{ij}) \quad (8)$$

$$\frac{\partial P_i}{\partial \delta_i} = -Q_{ii} - V_i^2 B_{ii} \quad (9)$$

Also, $J_{21} = \frac{\partial Q}{\partial \delta}$, where

$$\frac{\partial Q_i}{\partial \delta_j} = -V_i V_j Y_{ij} \cos(\delta_i + \delta_j - \theta_{ij}) \quad (10)$$

$$\frac{\partial Q_i}{\partial \delta_i} = P_i - V_i^2 G_{ii} \quad (11)$$

And $J_{12} = V * \frac{\partial P}{\partial V}$, where

$$V * \frac{\partial P_i}{\partial V_j} = V_i V_j Y_{ij} \cos(\delta_i + \delta_j - \theta_{ij}) \quad (12)$$

$$V * \frac{\partial P_i}{\partial V_i} = P_i + V_i^2 G_{ii} \quad (13)$$

Finally $J_{22} = V * \frac{\partial Q}{\partial V}$, where

$$V * \frac{\partial Q_i}{\partial \delta_j} = V_i V_j Y_{ij} \sin(\delta_i + \delta_j - \theta_{ij}) \quad (14)$$

$$V * \frac{\partial Q_i}{\partial V_i} = Q_i - V_i^2 B_{ii} \quad (15)$$

In power flow calculation, we first determine PV bus and PQ bus. Where, for PV bus, theta is calculated as an unknown quantity. For PQ bus, V and theta are calculated as unknowns. Of course, before calculating, we also need to determine whether PV bus Q is within limits. If Q exceeds limits, then we should convert PV bus to PQ bus for calculation. For a 14 buses system, there should be 13 buses left, excluding Slack bus.

And then compute the Jacobian matrix. Finally, we use the values of P and Q obtained to compare with the original P and Q to see if the results are within the error. If it's out of error, we do the next iteration.

3. The steps for iterations

At the outset, we assume that the initial V is 1, and the initial theta is 0 for the PV buses. Also, we assume that the initial theta is 0 for the PQ buses. The following is the details:

1	V	Vangle
2	1.06	0
3	1.045	0
4	1.01	0
5	1	0
6	1	0
7	1.07	0
8	1	0
9	1.09	0
10	1	0
11	1	0
12	1	0
13	1	0
14	1	0
15	1	0

Fig 2. The Initial Value for V and Theta

After that, we can calculate the Y bus from branch data. Then, we can follow the steps for Fig 3.

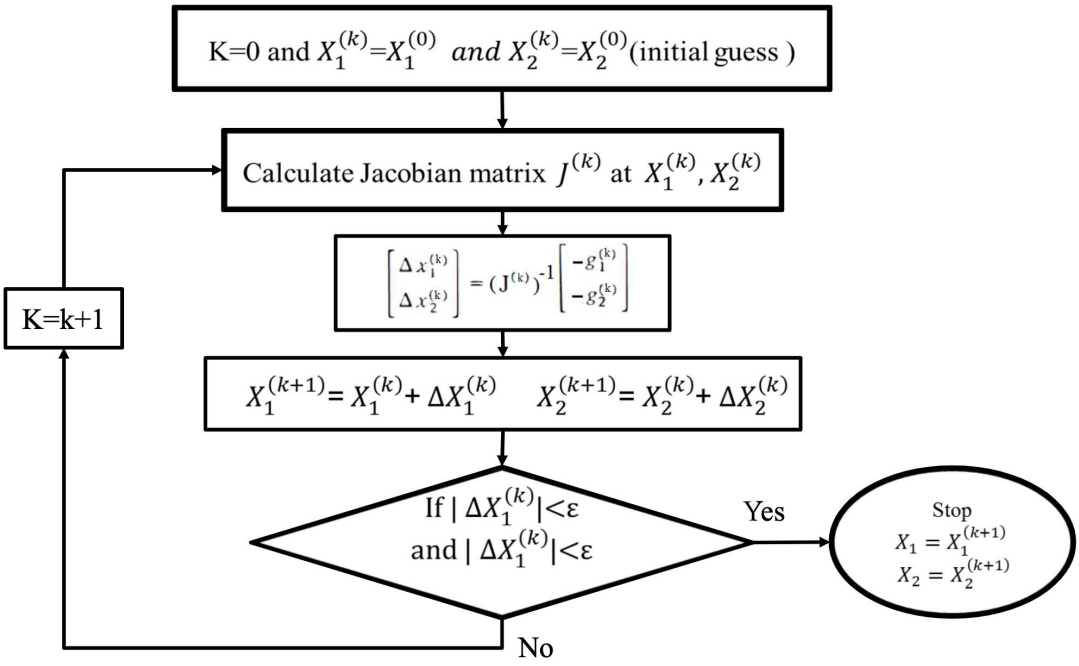

Fig 3. The Process for Newton-Raphson Power Flow

4. Results

Firstly, we can use branch data to get Y bus. The details is that Y_{ii} = Sum of admittance connected to node i, and Y_{ij} = - Sum of the admittance connected from node i to node j. The following is the result of our Matlab execution code calculation:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	6.0250 - 1...	-4.9991 + 1...	0.0000 + 0.00...	0.0000 + 0.00...	-1.0259 + 4.2...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
2	-4.9991 + ...	9.5213 - 30...	-1.1350 + 4.7...	-1.6860 + 5.1...	-1.7011 + 5.1...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
3	0.0000 + 0...	-1.1350 + 4...	3.1210 - 9.82...	-1.9860 + 5.0...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
4	0.0000 + 0...	-1.6860 + 5...	-1.9860 + 5.0...	10.5130 - 38....	-6.8410 + 21....	0.0000 + 0.00...	0.0000 + 4.8...	0.0000 + 0.0...	0.0000 + 1.85...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
5	-1.0259 + ...	-1.7011 + 5...	0.0000 + 0.00...	-6.8410 + 21....	9.5680 - 35.5...	0.0000 + 4.25...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
6	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 4.25...	6.5799 - 17.3...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	-1.9550 + 4....	-1.5260 + ...	-3.0989 + 6...	0.0000 + 0.0...
7	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 4.88...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 - 19....	0.0000 + 5.6...	0.0000 + 9.09...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
8	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 5.6...	0.0000 - 5.6...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
9	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 1.85...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 9.0...	0.0000 + 0.0...	5.3261 - 24.0...	-3.9020 + 1...	0.0000 + 0.0...	0.0000 + 0...	0.0000 + 0....	-1.4240 + 3....
10	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	-3.9020 + 10....	5.7829 - 14....	-1.8809 + 4....	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
11	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	-1.9550 + 4.0...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	-1.8809 + 4....	3.8359 - 8.4...	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.0...
12	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	-1.5260 + 3.1...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	4.0150 - 5....	-2.4890 + 2...	0.0000 + 0.0...
13	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	-3.0989 + 6.1...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	-2.4890 + ...	6.7249 - 10....	-1.1370 + 2....
14	0.0000 + 0...	0.0000 + 0....	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.00...	0.0000 + 0.0...	0.0000 + 0.0...	-1.4240 + 3.0...	0.0000 + 0.0...	0.0000 + 0.0...	0.0000 + 0...	-1.1370 + 2...	2.5610 - 5.3...

Fig 4. Y bus Matrix

Then, we used the equation 8,9,10,11,12,13,14,15 to calculate the Jacobian matrix. The following is the first time Jacobian matrix and the final converged Jacobian matrix:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
32.7276	-5.0470	-5.3461	-5.4277	0	0	0	0	0	0	0	0	0	-1.7619	-1.7777	0	0	0	0	0	0	0
-5.0470	10.1665	-5.1195	0	0	0	0	0	0	0	0	0	0	-2.0058	0	0	0	0	0	0	0	0
-5.3461	-5.1195	38.7891	-21.5786	0	-4.8895	0	-1.8555	0	0	0	0	0	10.4173	-6.8410	2.9940...	1.1362...	0	0	0	0	0
-5.4277	0	-21.5786	36.0508	-4.5555	0	0	0	0	0	0	0	0	-6.8410	9.4299	0	0	0	0	0	0	0
0	0	0	-4.5555	18.8644	0	0	0	0	-4.3807	-3.3983	-6.5299	0	0	2.7894e...	0	0	0	-2.09...	-1.6328	-3.3159	0
0	0	-4.8895	0	0	20.1675	-6.1879	-9.0901	0	0	0	0	0	2.9940...	0	2.4319...	5.5661...	0	0	0	0	0
0	0	0	0	0	-6.1879	6.1879	0	0	0	0	0	0	0	0	3.7890...	0	0	0	0	0	0
0	0	-1.8555	0	0	-9.0901	0	24.3400	-10.36...	0	0	0	-3.0291	1.1362...	0	5.5661...	5.3261	-3.9020	0	0	0	-1.4240
0	0	0	0	0	0	0	-10.3654	14.7683	-4.4029	0	0	0	0	0	0	-3.9020	5.7829	-1.88...	0	0	0
0	0	0	0	-4.3807	0	0	0	-4.4029	8.7836	0	0	0	0	0	0	0	-1.8809	3.6991	0	0	0
0	0	0	0	-3.3983	0	0	0	0	0	5.6503	-2.2520	0	0	0	0	0	0	0	3.9082	-2.4890	0
0	0	0	0	-6.5299	0	0	0	0	0	-2.2520	11.0969	-2.3150	0	0	0	0	0	0	-2.4890	6.5080	-1.1370
0	0	0	0	0	0	0	-3.0291	0	0	0	-2.3150	5.3440	0	0	0	-1.4240	0	0	0	-1.1370	2.5610
1.7619	2.0058	-10.6087	6.8410	0	-2.994...	0	-1.136...	0	0	0	0	0	38.5192	-21.5786	-4.8895	-1.8555	0	0	0	0	0
1.7777	0	6.8410	-9.7061	-2.789...	0	0	0	0	0	0	0	0	-21.5786	35.0165	0	0	0	0	0	0	0
0	0	-2.9940...	0	0	2.4319...	-3.789...	-5.566...	0	0	0	0	0	-4.8895	0	18.9305	-9.0901	0	0	0	0	0
0	0	-1.1362...	0	0	-5.566...	0	-5.3261	3.9020	0	0	0	1.4240	-1.8555	0	-9.0901	23.8450	-10.3...	0	0	0	-3.0291
0	0	0	0	0	0	0	3.9020	-5.7829	1.8809	0	0	0	0	0	0	-10.36...	14.7683	-4.40...	0	0	0
0	0	0	0	2.0919	0	0	0	1.8809	-3.9728	0	0	0	0	0	0	0	-4.4029	8.2104	0	0	0
0	0	0	0	1.6328	0	0	0	0	0	-4.1218	2.4890	0	0	0	0	0	0	0	5.2056	-2.2520	0
0	0	0	0	3.3159	0	0	0	0	0	2.4890	-6.9419	1.1370	0	0	0	0	0	0	-2.2520	10.2425	-2.3150
0	0	0	0	0	0	0	1.4240	0	0	0	1.1370	-2.5610	0	0	0	-3.0291	0	0	0	-2.3150	5.3440

Fig 5. The first Jacobian matrix

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
32.8588	-5.1614	-5.6202	-5.68...	0	0	0	0	0	0	0	0	0	-1.2533	-1.4112	0	0	0	0	0	0	0
-4.8433	10.0016	-5.1583	0	0	0	0	0	0	0	0	0	0	-2.2056	0	0	0	0	0	0	0	0
-5.2821	-5.3199	40.4444	-22.4...	0	-5.3467	0	-2.0135	0	0	0	0	0	10.3170	-7.5999	0.2776	0.1619	0	0	0	0	0
-5.4369	0	-22.8...	37.35...	-4.6555	0	0	0	0	0	0	0	0	-6.4218	9.7834	0	0	0	0	0	0	0
0	0	0	-4.65...	19.9072	0	0	0	0	-4.6809	-3.6259	-6.9450	0	0	-0.4260	0	0	0	-2.0361	-1.5782	-3.1979	0
0	0	-5.3467	0	0	22.2961	-6.6171	-10.3323	0	0	0	0	0	-0.2899	0	0.0107	0.2900	0	0	0	0	0
0	0	0	0	0	-6.6171	6.6171	0	0	0	0	0	0	0	0	-9.950...	0	0	0	0	0	0
0	0	-2.0135	0	0	-10.33...	0	27.3936	-11.66...	0	0	0	-3.3818	-0.1681	0	-0.2884	5.3503	-4.1255	0	0	0	-1.4535
0	0	0	0	0	0	0	-11.6469	16.5752	-4.9283	0	0	0	0	0	0	-4.1502	6.0219	-2.0267	0	0	0
0	0	0	0	-4.6244	0	0	0	-4.9626	9.5870	0	0	0	0	0	0	0	-1.9598	4.0502	0	0	0
0	0	0	0	-3.5705	0	0	0	0	0	6.0898	-2.5193	0	0	0	0	0	0	0	4.1975	-2.6311	0
0	0	0	0	-6.8192	0	0	0	0	0	-2.5083	11.8912	-2.5637	0	0	0	0	0	0	-2.6284	6.9679	-1.1509
0	0	0	0	0	0	0	-3.3226	0	0	0	-2.5152	5.8378	0	0	0	-1.5412	0	0	0	-1.2301	2.5111
2.3095	1.8465	-11.4...	7.8010	0	-0.2969	0	-0.1722	0	0	0	0	0	39.6857	-21.90...	-4.9999	-1.8934	0	0	0	0	0
2.1928	0	6.5769	-10.1...	-0.4373	0	0	0	0	0	0	0	0	-22.33...	36.5560	0	0	0	0	0	0	0
0	0	0.2969	0	0	0.0115	-1.874...	-0.3084	0	0	0	0	0	-5.2206	0	20.9598	-9.7162	0	0	0	0	0
0	0	0.1722	0	0	0.3084	0	-6.3563	4.3628	0	0	0	1.5129	-1.9660	0	-9.6622	25.4801	-11.0316	0	0	0	-3.2490
0	0	0	0	0	0	0	4.4134	-6.5661	2.1527	0	0	0	0	0	0	-10.9525	15.5613	-4.6399	0	0	0
0	0	0	0	2.2809	0	0	0	2.0725	-4.3534	0	0	0	0	0	0	0	-4.6927	9.0247	0	0	0
0	0	0	0	1.7865	0	0	0	0	0	-4.5599	2.7734	0	0	0	0	0	0	0	5.7453	-2.3900	0
0	0	0	0	3.6185	0	0	0	0	0	2.7833	-7.5997	1.1979	0	0	0	0	0	0	-2.3686	11.2130	-2.4630
0	0	0	0	0	0	0	1.6389	0	0	0	1.2966	-2.9355	0	0	0	-3.1245	0	0	0	-2.3861	5.5164

Fig 6. The final converged Jacobian matrix

Finally, we can get the final result for voltage and voltage angle. The following is the result, and the mismatches is smaller than 0.01.

V	Vangle
1.06	0
1.045	-4.9807
1.01	-12.722
1.0177	-10.31
1.0196	-8.7713
1.07	-14.216
1.0616	-13.358
1.09	-13.358
1.056	-14.938
1.051	-15.096
1.0569	-14.787
1.0552	-15.071
1.0504	-15.152
1.0356	-16.032

Fig 7. The final Voltage and Voltage angle

Then we compare the calculated results and the given solution.

Given Solution		Calculated Results	
Bus Voltage (pu)	Delta (in degrees)	V(pu)	Vangle(in degrees)
1.06	0	1.06	0
1.045	-4.98	1.045	-4.980672402
1.01	-12.73	1.01	-12.72161819
1.018	-10.31	1.017723715	-10.31038324
1.02	-8.77	1.019569292	-8.771304576
1.07	-14.22	1.07	-14.21617124
1.062	-13.36	1.061584742	-13.35799983
1.09	-13.36	1.09	-13.35799983
1.056	-14.94	1.055995799	-14.9382869
1.051	-15.1	1.0510419	-15.09632002
1.057	-14.79	1.056949103	-14.78729493
1.055	-15.08	1.055207548	-15.07106272
1.05	-15.16	1.050404666	-15.15213035
1.036	-16.03	1.035578231	-16.0324032

Fig 8. Comparison of results

The results show that Newton Raphson's method is very accurate for power flow. In addition, this method is very fast. The iteration is made only twice, with the mismatches of less than 0.01.

5.Appendix

```
clc
clear all
%  Brahcn    Bus    To bus  R          X          Line          Rtio
%                                                Charging B
linedata=[   1      2       0.01938    0.05917    0.0528         1
             1      5       0.05403    0.22304    0.0492         1
             2      3       0.04699    0.19797    0.0438         1
             2      4       0.05811    0.17632    0.0340         1
             2      5       0.05695    0.17388    0.0346         1
             3      4       0.06701    0.17103    0.0128         1
             4      5       0.01335    0.04211    0.0            1
             4      7       0.0        0.20912    0.0        0.978
```

```
    4     9     0.0      0.55618   0.0      0.969
    5     6     0.0      0.25202   0.0      0.932
    6     11    0.09498  0.19890   0.0         1
    6     12    0.12291  0.25581   0.0         1
    6     13    0.06615  0.13027   0.0         1
    7     8     0.0      0.17615   0.0         1
    7     9     0.0      0.11001   0.0         1
    9     10    0.03181  0.08450   0.0         1
    9     14    0.12711  0.27038   0.0         1
   10     11    0.08205  0.19207   0.0         1
   12     13    0.22092  0.19988   0.0         1
   13     14    0.17093  0.34802   0.0         1];

numberline=length(linedata(:,1))
a=linedata(:,1);              % Number of Buses
b=linedata(:,2);              % Number of to Buses
R=linedata(:,3);              % Get the resistance
X=linedata(:,4);              % Get the Reactance
B_Charging=i*linedata(:,5)/2;          % Get B/2
T=linedata(:,6);              % Get the ratio of transformer
Ti=T*i;


   %      | 1   | 2  | 3      |   4         |    5    |    6     |   7   |     8      |   9    |      10
   |    11    |    12    |    13    |    14   |   15    |
    %     |  Bi   | Type| Fin-V   | FinAng-Deg |   PL-MW   | QL-MVAR   |   PGen | QGen-MVAR |
BaseKV | DesiredVolts | MaxMVAR | MinMVAR | ShuntG |   ShuntB| Remote |

node    =
[  1     3    1.060      0.0       0.0       0.0      232.4    -16.9     0.0     1.060        0.0       0.0      0.0      0.0      0;
           2    2   1.045     -4.98     21.7     12.7     40.0     42.4     0.0     1.045      50.0     -40.0     0.0      0.0      0;
           3    2   1.010    -12.72     94.2     19.0      0.0     23.4     0.0     1.010      40.0      0.0      0.0      0.0      0;
           4    0   1.019    -10.33     47.8     -3.9      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
           5    0   1.020     -8.78      7.6      1.6      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
           6    2   1.070    -14.22     11.2      7.5      0.0     12.2     0.0     1.070      24.0      -6.0      0.0      0.0      0;
           7    0   1.062    -13.37      0.0      0.0      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
           8    2   1.090    -13.36      0.0      0.0      0.0     17.4     0.0     1.090      24.0      -6.0      0.0      0.0      0;
           9    0   1.056    -14.94     29.5     16.6      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.19     0;
          10    0   1.051    -15.10      9.0      5.8      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
          11    0   1.057    -14.79      3.5      1.8      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
          12    0   1.055    -15.07      6.1      1.6      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
          13    0   1.050    -15.16     13.5      5.8      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0;
          14    0   1.036    -16.04     14.9      5.0      0.0      0.0     0.0      0.0        0.0       0.0      0.0      0.0      0];
```

```
numberbus=max(node(:,1));            % Get the number of the node
V=node(:,10);
V(~V)=1;                             % The initial voltage from 1
Vangle=zeros(numberbus,1);           % The initial angle of voltage from 0
P0=(node(:,7)-node(:,5))/100;            % Get P of each bus
Q0=(node(:,8)-node(:,6))/100;            % Get Q of each bus
C=node(:,13)+i*node(:,14);               % Get the Shunt

PVbus=node(:,2)==2;                      % Define 2 is PV bus
PQbus=node(:,2)==0;                      % Define 0 is PQ bus
numberPVbus=sum(PVbus);                      % Define The number of PV bus
numberPQbus=sum(PQbus);                      % Define the number of PQ bus
PQ=find(node(:,2)==0|node(:,2)==1);      % Find the order of PQ from the form


  Z=R+i*X;          % Define the impedance

  Y=(1./Z);         % Define the resistance


  %=====================off Diagonal for Y bus=============================
  Ybus=zeros(14,14);       % Create a new empty Y matrix bus
  for k=1:numberline
   Ybus(linedata(k,1),linedata(k,2))=Ybus(linedata(k,1),linedata(k,2))-Y(k)/T(k);
   Ybus(linedata(k,2),linedata(k,1))=Ybus(linedata(k,1),linedata(k,2));
  end

  %===================Diagonal for Y bus============================

 for m=1:numberbus
     for n=1:numberline
         if m==linedata(n,1);
             Ybus(m,m)=Ybus(m,m)+Y(n)/(T(n)^2)+B_Charging(n);
         elseif m==linedata(n,2);
             Ybus(m,m)=Ybus(m,m)+Y(n)+B_Charging(n);
         end
     end
     Ybus(m, m) = Ybus(m, m) + C(m);
 end


Yabs=abs(Ybus);      % Get the real of Y bus
Yangle=angle(Ybus);      % Get the angle of Y bus
Gi=real(Ybus);               % Get the real of Y bus
```

```
    Bi=imag(Ybus);            % Get the imag of Y bus

%%%%%%%%%%% Get the initial P and Q Value %%%%%%%%%%%%%%%%%%%%%%%%%%%
      Pi=zeros(numberbus,1);
      Qi=zeros(numberbus,1);
      for m=1:numberbus
          for n=1:numberbus
              Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
              Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

          end

      end

      dP=P0-Pi;
      dQ=Q0-Qi;
      dm=[dP(2:end);dQ(PQ)];   % Get the first time mismatches

      Tol=1
      Iter=0;

 while max(abs(Tol))>1e-2 && (Iter<5)



      %%%%%%%%% Jacobian %%%%%%%%%
      %%%%%%%%% J11 %%%%%%%%%%%
      for m=1:(numberPQbus+numberPVbus)
          for n=1:(numberPQbus+numberPVbus)
      if m==n
          J11(m,n)=-Qi(m+1)-Bi(m+1,n+1)*V(m+1)^2;
      else
          J11(m,n)=V(m+1)*V(n+1)*Yabs(m+1,n+1)*sin(Vangle(m+1)-Vangle(n+1)-Yangle(m+1,n+1)
);
      end
          end
      end

      %%%%%%%%% J12 %%%%%%%%%%%
      for m=2:(numberPQbus+numberPVbus+1)
          for n=2:numberPQbus+1
              k=PQ(n-1);
              if k==m
          J12(m-1,n-1)=Pi(k)/V(k)+Gi(m,k)*V(k);
```

```
else
    J12(m-1,n-1)=V(m)*Yabs(m,k)*cos(Vangle(m)-Vangle(k)-Yangle(m,k));
end
    end
end

%%%%%%%% J21 %%%%%%%%%%
for m=2:numberPQbus+1
    k=PQ(m-1)
    for n=2:(numberPQbus+numberPVbus+1)
        if n==k
    J21(m-1,n-1)=Pi(k)-Gi(k,n)*V(k)^2;
else
    J21(m-1,n-1)=-1*V(k)*V(n)*Yabs(k,n)*cos(Vangle(k)-Vangle(n)-Yangle(k,n));
end
    end
end

%%%%%%%% J22 %%%%%%%%%%
for m=2:numberPQbus+1
    k1=PQ(m-1)
    for n=2:numberPQbus+1
        k2=PQ(n-1)
        if k1==k2
    J22(m-1,n-1)=Qi(k1)/V(k1)-Bi(k1,k2)*V(k1);
else
    J22(m-1,n-1)=V(k1)*Yabs(k1,k2)*sin(Vangle(k1)-Vangle(k2)-Yangle(k1,k2));
end
    end
end

J=[J11 J12; J21 J22];


X0=[Vangle(2:end);V(PQ)];   % Create a vector that made of Vangle and V
X=X0+inv(J)*(dm);

Vangle0=X(1:numberbus-1);   % Get the Vangle after the calculation
V0=X(numberbus:end);        % Get the V after the calculation


Vangle=[0;Vangle0];     % Get the Vagnle excepting slack bus
V=node(:,10);               % Get the V from the node data
V(~V)=V0;                   % Use the V0 to instead of 0
```

```
        %%%%%%   Get a new P and Q after the calculation %%%%%%%%%%%%%
         Pi=zeros(numberbus,1);
         Qi=zeros(numberbus,1);
        for m=1:numberbus
            for n=1:numberbus
                Pi(m)=Pi(m)+Yabs(m,n)*V(m)*V(n)*cos(Vangle(m)-Vangle(n)-Yangle(m,n));
                Qi(m)=Qi(m)+Yabs(m,n)*V(m)*V(n)*sin(Vangle(m)-Vangle(n)-Yangle(m,n));

            end

        end

        dP=P0-Pi;
        dQ=Q0-Qi;
        dm=[dP(2:end);dQ(PQ)];   % Mismatches

        Tol=dm;
        Iter=Iter+1;

 end
Vangle=Vangle*(180/pi);
```