MIDDLE EAST TECHNICAL UNIVERSITY ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT

EE-472 Power System Analysis II

Spring 2024 Project-1

Canberk Kaçan

01.06.2024

Introduction

In the operation of electrical power systems, load flow analysis is a fundamental analysis for ensuring the safe operation and planning of power networks. The Newton-Raphson Load Flow Analysis is a widely utilized method due to its ease of convergence and robustness in handling non-linear equations. This project aims to develop a generic Python function capable of performing Newton-Raphson Load Flow Analysis while incorporating generator reactive power limit considerations. The implementation takes place in Python 3.12 environment and is designed to be adaptable to various power systems that are presented in a common IEEE data format, such as the IEEE 30-bus and IEEE 118-bus systems. Moreover, by considering the reactive power limits of generators, the function ensures a more realistic and accurate representation of the power network's operating conditions. Consideration of reactive power limitations is crucial for preventing voltage instability and maintaining system reliability along with the bus voltages.

The Project

The Python function I have implemented takes the file path of the CDF file and gives the calculated bus voltages, bus angles, computation time, iteration number, power losses in the system and PV bus numbers that have stayed in the Q-limits.

First, the Y_{BUS} is constructed using bus and line data supplied by the CDF file. Then, necessary vectors are constructed such as voltage vectors for the flat start. Due to the flat start assumption, the initial voltages and angles of the buses are assumed to be 1 p.u. and 0 degrees respectively. After the flat start adjustments, a mismatch vector is formed by differencing the calculated and given power values of the buses. Jacobian is formed after the mismatch vector. With Jacobian being constructed, the function starts to Newton-Raphson iteration since all the necessary parameters are known: initial bus angles, initial PQ bus voltages, calculated bus powers and Y_{BUS}. The algorithm stops when a certain convergence is met. For the operation, the minimum convergence of the iterations is determined as 0.01 which gives the end values a maximum error percentage of 1%. After the convergence is met the function specifies the PV buses that did not stick to the respective Q limits. A block diagram showing the fundamental operating principle of the algorithm is given in Figure 1.

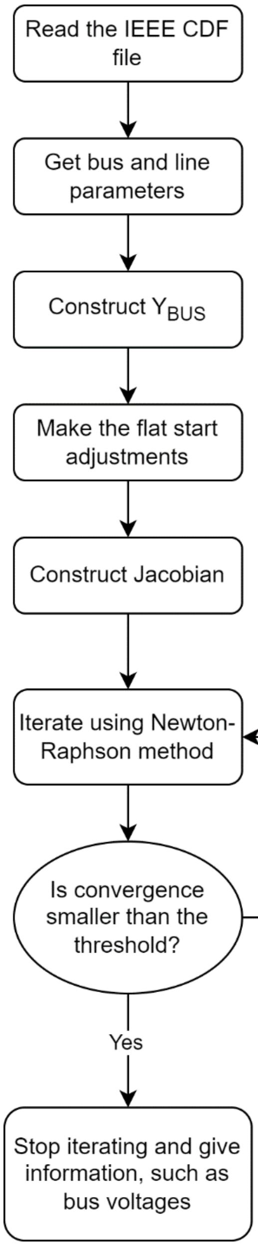

Figure 1. Block diagram of the algorithm

The function also saves the final bus voltages, final bus angles and PV buses that were not stuck to the Q limits to Excel files for the ease of user. The bus voltages and angles are given with their respective bus numbers. Moreover, at the end, the function prints final bus voltages in p.u., bus angles in degrees, solution time in hh:mm:ss format, number of iterations, active losses in the system along with the reactive losses in the system and numbers of the PV buses that did not stick to their respective Q limits. The function also writes bus voltages, bus angles and PV buses that stick to their limits to Excel files to make reading the values easier.

Convergence threshold and assumptions

The convergence threshold (ε) was chosen as 0.01. The reason for this choice is to keep the computation time small while having a reasonable convergence.

There were three main assumptions I have made:

- The slack bus and PV bus voltages were always kept at 1 p.u..
- The slack bus angle is always 0 degrees.
- The PV buses supply the exact P values that are given in CDF.

Test Results

The test results given below are the outputs of the implemented power flow solution and it should be noted that the results are based on the IEEE 300 bus system.

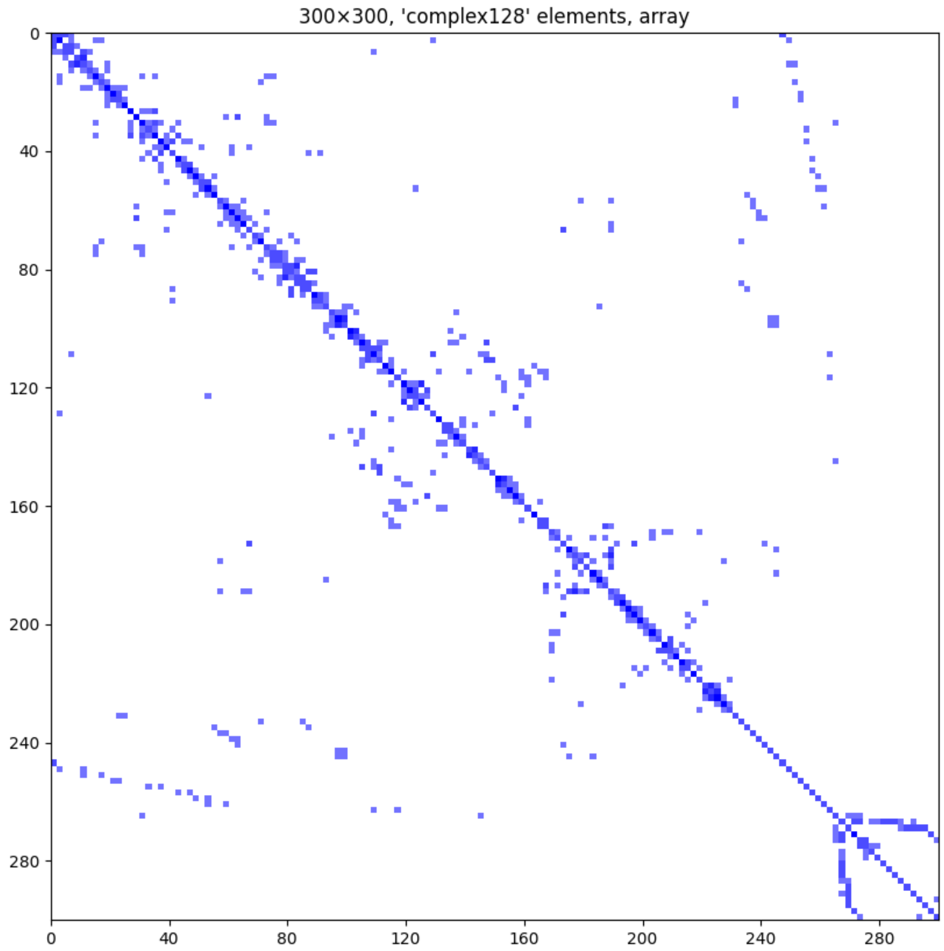

Figure 2. IEEE 300 bus system, sparsity plot of the admittance matrix

Table 1. Buses and their respective final voltages and angles

Bus Numbers	Bus Voltages (p.u.)	Bus Angles (Degrees)
1	1.000922	27.54901
2	1.008224	29.32434

3	0.98892	28.12708
4	1.028565	25.86464
5	0.995164	26.19958
6	1.007437	28.51498
7	0.984373	27.74349
8	1	23.43937
9	0.985012	24.28621
10	1	22.78748
11	0.992039	23.84979
12	0.984822	26.76818
13	0.991426	20.90259
14	0.990304	15.31371
15	1.021397	10.58447
16	1.036156	16.86671
17	1.026055	6.601909
19	0.981222	22.44418
20	1	19.09332
21	0.97139	23.15152
22	0.989263	19.67615
23	1.017047	26.15752
24	0.990777	27.85933
25	0.99613	23.42273
26	0.979245	20.0764
27	0.964624	16.89058
33	1.033957	3.303234
34	1.047975	9.492985
35	1.005959	353.4031
36	1.024841	356.3658
37	1.023086	3.140558
38	1.029162	2.594237
39	1.052762	12.10921
40	1.029502	2.463505
41	1.038179	4.322007
42	1.051286	10.26755
43	1.017201	359.8877
44	1.031944	0.313385
45	1.046124	3.526292
46	1.050682	6.313071
47	1.007126	355.1294
48	1.003545	0.995603
49	1.000412	0.333293
51	1.014133	2.262708
52	0.997837	1.691289
53	1.012914	359.1303
54	1.022195	0.593049

55	1.021381	3.69929
57	1.028814	11.55977
58	0.991991	14.48116
59	0.979352	15.43259
60	1.052449	9.897418
61	0.989998	17.32124
62	1.028276	20.18897
63	1	4.41274
64	0.989955	8.674794
69	0.994174	353.8849
70	0.986114	344.6163
71	1.00673	349.5504
72	0.997553	351.8261
73	1.006907	353.2057
74	1.024287	356.7534
76	1	352.4758
77	1.005147	354.3663
78	1.006913	355.277
79	1.003258	354.3119
80	1.003103	354.5082
81	1.039802	0.234128
84	1	2.248721
85	0.996127	1.552633
86	0.995577	5.409268
87	0.995573	12.67202
88	1.033417	358.0151
89	1.002705	7.274299
90	1.011966	5.603312
91	1	10.74016
92	1	12.93291
94	0.989037	10.77792
97	0.998914	6.313011
98	1	4.681676
99	0.999699	358.7076
100	0.999045	4.956683
102	0.998251	4.157445
103	1.002293	6.89821
104	0.998325	1.907704
105	1.001633	5.686046
107	1.007358	2.018887
108	1	358.7316
109	0.998877	352.9214
110	0.991912	354.2719
112	1.005531	350.5292
113	0.998578	353.1888

114	1.008791	350.629
115	0.966632	9.660357
116	1.004357	10.81245
117	0.934563	18.3556
118	0.928699	18.88998
119	1	28.12265
120	0.953294	14.43382
121	0.993896	10.43385
122	0.978073	9.001082
123	0.989041	5.792623
124	1	10.06559
125	1	4.922071
126	1.00105	10.46532
127	0.998593	12.83824
128	1.00645	18.33664
129	1.00807	18.68609
130	1.030088	27.75369
131	0.983133	27.75366
132	1.035102	25.1829
133	1.000369	17.84809
134	1.002811	15.7276
135	0.99859	17.23026
136	1.013343	26.3943
137	1.001956	22.74473
138	1	17.38012
139	0.969208	20.48593
140	0.99826	20.62135
141	1	24.67383
142	0.981589	21.11323
143	1	28.00473
144	1.024785	20.81335
145	0.971756	23.99461
146	1	29.1112
147	1	33.6273
148	1.022498	24.18327
149	1	29.82847
150	0.985633	28.02327
151	1.034745	26.18727
152	1	35.21372
153	1	36.71895
154	0.982793	24.73169
155	1.04358	32.03729
156	1	30.46043
157	0.989221	11.44175
158	0.997858	12.01176

159	0.990348	13.55429
160	0.987353	10.98549
161	1.016257	34.73138
162	1.027633	43.64266
163	1.031247	27.77521
164	1.010732	34.92507
165	1.023192	51.46597
166	1.006375	55.37987
167	0.982832	16.13891
168	1.00642	18.31569
169	0.993248	16.49997
170	1	22.36255
171	1	13.42792
172	0.987114	17.89558
173	0.958293	11.67321
174	1.019488	21.77214
175	0.944805	16.92316
176	1	30.05368
177	1	24.83605
178	0.922435	17.87778
179	0.947525	15.01582
180	0.951687	21.13178
181	1.001222	22.94285
182	0.999743	19.81396
183	0.946881	33.40506
184	0.999691	17.14
185	1	19.97491
186	1	26.89893
187	1	26.04643
188	1.000116	23.61646
189	1.0273	353.7683
190	1	359.8707
191	1	32.36242
192	0.973927	9.626421
193	1.048471	353.093
194	1.033102	0.577711
195	1.025176	358.902
196	1.044268	353.197
197	1.01831	355.718
198	1	359.5021
199	0.982603	353.6867
200	0.983689	353.9069
201	1.015497	352.4132
202	0.995603	354.3974
203	0.998195	357.555

204	1.04462	353.6619
205	1.048004	353.2121
206	1.044275	351.9497
207	1.055123	351.9218
208	1.035403	353.1615
209	1.019875	354.276
210	1.00777	355.3123
211	1.008478	356.4156
212	1.015647	357.2703
213	1	8.398605
214	0.995482	2.467912
215	0.987403	359.7881
216	0.968867	357.6342
217	0.998004	357.9806
218	0.98378	357.5503
219	1.01922	359.0415
220	1	358.5487
221	1	357.9244
222	1	356.905
223	0.996148	357.7276
224	0.993012	358.9959
225	0.979632	9.408385
226	1.000645	358.9477
227	1	352.7232
228	1.007468	359.4497
229	1.012747	0.58071
230	1	7.268177
231	1.010997	359.0687
232	1.004399	356.8956
233	1	354.0109
234	1.015336	359.4446
235	0.999195	359.3087
236	1	5.237871
237	1.014912	359.1805
238	1	359.3924
239	1	4.504472
240	0.971977	0.392252
241	1	4.290918
242	1	2.461468
243	1	1.148042
244	0.986889	0.126554
245	0.974712	359.3003
246	0.966659	358.539
247	0.979035	358.48
248	0.995805	354.8298

249	0.996711	354.4243
250	1.042448	356.2714
281	0.973359	0.468674
319	1.00358	23.22431
320	0.995495	19.57012
322	1.00304	1.61985
323	0.985897	5.89478
324	0.986851	355.6835
526	1.023866	349.543
528	1.007545	342.3784
531	0.989144	350.3147
552	1.022841	355.9193
562	1.014035	350.8948
609	0.982984	350.7383
664	1.015186	2.970336
1190	0.971105	26.78836
1200	1.020539	15.65727
1201	0.990166	7.7532
2040	1.057849	4.839874
7001	1	32.77467
7002	1	34.59258
7003	1	35.43424
7011	1	26.428
7012	1	33.41816
7017	1	9.547843
7023	1	28.59545
7024	1	34.64831
7039	1	20.84233
7044	1	3.944107
7049	1	0
7055	1	8.461375
7057	1	16.36198
7061	1	22.7758
7062	1	27.18662
7071	1	353.9343
7130	1	42.04073
7139	1	27.1838
7166	1	60.25929
9001	1.014716	3.144417
9002	1	355.5651
9003	1.010106	354.7032
9004	1.003897	354.5628
9005	1.015455	3.251116
9006	1.025681	356.9521
9007	1.016422	355.688

9012	1.009187	357.1229
9021	0.994399	355.3454
9022	0.972066	352.8893
9023	0.981145	355.0293
9024	0.977475	353.0873
9025	0.971708	354.0309
9026	0.972805	354.1156
9031	0.965712	349.7975
9032	0.975688	350.8383
9033	0.963673	349.5255
9034	1.02564	353.3058
9035	0.980064	351.4256
9036	0.988942	351.8871
9037	0.986264	351.9605
9038	0.971274	350.3177
9041	0.991394	353.146
9042	0.979331	352.0827
9043	0.992553	353.0516
9044	1.00599	354.6035
9051	1	355.1434
9052	0.989904	357.3974
9053	1	17.62028
9054	1	7.795016
9055	1	7.077558
9071	1.001316	353.9574
9072	1.005828	354.4892
9121	0.987945	355.1349
9533	1.847456	54.6141

Table 2. PV buses that stuck to Q limits

Bus Count	Bus Number
0	63
1	84
2	91
3	92
4	98
5	108
6	119
7	124
8	125
9	138
10	141

11	143
12	146
13	147
14	149
15	170
16	171
17	176
18	185
19	186
20	187
21	191
22	198
23	213
24	221
25	222
26	227
27	230
28	233
29	238
30	239
31	241
32	243
33	7001
34	7002
35	7003
36	7011
37	7012
38	7017
39	7024
40	7039
41	7044
42	7055
43	7057
44	7061
45	7062
46	7071
47	7130
48	7139
49	7166
50	9051
51	9054
52	9055

Computational Methods for Improved Performance, and Hardware Specifications

Several techniques were used to improve the computation performance.

- Use of arrays instead of lists to prevent dynamic memory allocation.
- Precreating arrays with their final sizes to prevent dynamic array resizing.
- Adding a breakpoint to prevent an infinite loop in case of not finding a convergent solution.

The average solution time for the 300 bus system is 1 second.

Hardware:

- Processor: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz
- RAM: 8GB
- Windows 10-64 byte operating system
- Brand: MSI
- Model: GL73 8RD

Requirements

- Python 3.12 environment
- Pandas library
- Numpy library
- Math library
- Matspy library
- Datetime library
- IEEE CDF file
- IEEE CDF file path

Additional Remarks

In the case of a non-convergent case, I have added a so-called "choose the best algorithm". If 250 iterations are made and the solution cannot be in the convergence threshold, the function prints that the solution cannot be reached; however, it outputs the voltage values which had the lowest mismatch values among the iterations.

Moreover, writing the outputs to Excel can cause problems sometimes, if this happens which means an error regarding pandas, just delete the already constructed Excel files and run the script again.