Phase 1 Due: 01.12.2025 23:59
Phase 2 Due: 05.01.2025 23:59

EE-471 Power System Analysis I
2024-2025 Fall Project

Aim: To implement a function in PYTHON 3.13.X environment that can perform Newton-Raphson Load Flow Analysis. The functions should be generic, meaning that they should be capable of solving the problem for different systems given with the same input format. The project consists of two phases.

Phase 1: Implement a function named "create_bus_admittance_matrix(*args, **kwargs)". The function should read the input file and return the following variables in the following order.

1. Bus admittance matrix of the system (y_bus).
   Explanation: Should be in scipy lil_matrix format where the size of the matrix is equal to the number of buses. The rows and columns should be ordered according to the appearance order in "bus_data" dictionary of the input file.

Phase 2: Implement a function named "load_flow(*args, **kwargs)". The function should take the bus admittance matrix as input and return the following variables in the following order.

1. Voltage magnitude values at each bus in per unit (voltage_magnitude).
   Explanation: Dictionary where keys are bus IDs and values are voltage magnitudes in per unit.

2. Voltage angle values at each bus in degrees (voltage_angle).
   Explanation: Dictionary where keys are bus IDs and values are voltage angles in degrees.

3. Real power losses in the system (p_loss).
   Explanation: Float in MW.

4. Reactive power losses in the system(q_loss).
   Explanation: Float in MVAR

Include a 'requirements.txt' file in your repository for any dependencies your project has.

Submission: Submit the final version of your report via ODTUClass to "Project" assignment. Commit and push the final version of your codes and work to your GitHub repository. The report should **NOT** include detailed explanation of your code; rather it should include the following.

a) Results of load flow analysis. Create tables for each input file. Compare your results with PSSE (for all cases) and DIgSilent (cases 14 bus and 30 bus).
b) Convergence threshold. The change of solution time and solution error vs convergence threshold.
c) Any computational method you used to improve computational performance.
d) Flow charts of the code you implemented.
e) Explanation of the solution.
f) Any additional reasoning and comments.
g) Sparsity pattern plot of Y_{BUS}.

Bonus 10 points will be awarded to the fastest running code (average of 100 runs).

Note 1: You are responsible whether your project is running or not. Test cases will not be provided. Use PSSE and DigSilent to test and validate your projects.

Note 2: Follow the variable name conventions for outputs.

For your questions related to the project use "Project" discussion under "Anouncements" in ODTUClass, so that all can benefit from the answers.