McGill University

ECSE 563 Power System Operation and Planning—Fall 2025

Assignment 1
Network Analysis

Complete the following function programming assignments. In all cases, provide fully-documented code with appropriate software validation as instructed. Note that if you are using Matlab (or another adequate programming language) to perform basic matrix operations, all of your programs should contain fewer than 25 lines of code.

Throughout this assignment, to mirror how practical power systems are handled, you are forbidden from using a full matrix inversion function. If using Matlab, rely on functions like `linsolve()` or the backslash operator.

Submit a report in .pdf demonstrating your development process and your validation evidence. In your report, provide a link to an online repository to access your code (e.g., github, OneDrive, etc.)

1. Develop a callable function to calculate the admittance matrix of any ac network. The module should be able to take as inputs vectors `nfrom`, `nto` describing the connectivity of the network, and vectors `r`, `x` and `b` providing respectively the line resistance, reactance and shunt susceptance. The output should be the admittance matrix, `Y`, of appropriate dimensions; use the following function prototype:

   ```
   Y = admittance(nfrom, nto, r, x, b)
   ```

   Provide fully-documented code with appropriate validation using the IEEE 9-bus test system found on myCourses (`ieee9_A1.m`).

   Using the internal current source values provided, determine the voltages at each node of the network. Express your results in polar coordinates.

2. Develop a callable function to calculate the impedance matrix of any network. The module should be able to take as inputs vectors `nfrom`, `nto` describing the connectivity of the network, and vectors `r`, `x` and `b` providing respectively the line resistance, reactance and shunt susceptance. The output should be the impedance matrix, `Z`, of appropriate dimensions; use the following function prototype:

   ```
   Z = impedance(nfrom, nto, r, x, b)
   ```

   Provide fully-documented code with appropriate validation using the IEEE 9-bus test system. *Hint:* Reuse functions you have developed previously.

3. Develop a callable function to calculate the fault currents and values of node voltages during balanced three-phase faults. The module should be able to take as inputs `Y`, the admittance matrix of the healthy network, `Iint`, the vector of pre-fault internal currents, `idfault`, the index of the faulted node and `Zf`, the impedance of the fault. The output should be `If`, the fault current at `idfault` and `Vf`, the vector of network node voltages during the fault; use the following function prototype:

```
[If, Vf] = fault(Y, Iint, idfault, Zf)
```

   Validate your software by calculating the bolted fault currents (*i.e.* $Z_f = 0$) at each node for the IEEE 9-bus test system. At the same time, monitor the voltage magnitudes at the other nodes. Discuss your results.

4. Develop a callable function to calculate the generalized Thévenin equivalent of a network. The module should be able to take as inputs `Y`, the admittance matrix of the network, `Iint`, the vector of pre-fault internal current sources and `id`, the vector of network node indices where the generalized Thévenin equivalent should be evaluated. The output should be `Eeq`, the vector of generalized Thévenin equivalent voltages and `Zeq`, the generalized Thévenin equivalent impedance matrix; use the following function prototype:

```
[Eeq, Zeq] = genthevenin(Y, Iint, id)
```

   (a) Validate your software by calculating the generalized Thévenin equivalent of the IEEE 9-bus test system simultaneously looking into nodes 3 and 5. Use the set of pre-fault internal current values as provided in the IEEE 9-bus test system.

   (b) Calculate the generalized Thévenin equivalent simultaneously looking into nodes 9 and 4. Provide a sketch of the resulting Thévenin equivalent circuit.

5. Develop a callable function to perform generalized fault calculations. The module should be able to take as inputs `YN`, the admittance matrix of the healthy network, `YF`, the admittance matrix of the fault network, `IintN`, the vector of pre-fault internal current sources in the healthy network, `IintF`, the vector of pre-fault internal current sources in the fault network, `idN`, the vector of network node indices in the healthy network to which the fault network is connected and `idF`, the vector of network node indices in the fault network to which the healthy network is connected. The output should be `IT`, the vector of tie line currents between the healthy and fault network and `VNF` the vector of node voltages in the healthy network with the fault network connected; use the following function prototype:

```
[IT, VNF] = genfault(YN, YF, IintN, IintF, idN, idF)
```

(a) Use your module to show the impact of an outage of line 8 (which links nodes 8 and 9) of the IEEE 9-bus test system. Discuss your results.

(b) Validate your software by calculating the tie line current and healthy network voltages if two of the IEEE 9-bus test system are connected together at node 1 for `YN` and node 5 for `YF`. Use the set of pre-fault internal current provided for both networks.

(c) Repeat the calculation this time connecting node 3 (healthy network) to node 7 (fault network) and node 5 (healthy network) to node 4 (fault network).

(d) Finally, you are to interconnect the IEEE 24-bus test system found on myCourses (`ieee24_A1.m`) with itself between nodes 7 (of the first network) and 3 (of the second network), 13 (of the first network) and 15 (of the second network), and 23 (of the first network) and 17 (of the second network). Determine voltages at all nodes (in the first system) after interconnecting them. What are the tie line currents?

Appendix

The IEEE 9-bus and 24-bus test networks data files are available on myCourses (ieee9_A1.m and ieee24_A1.m).