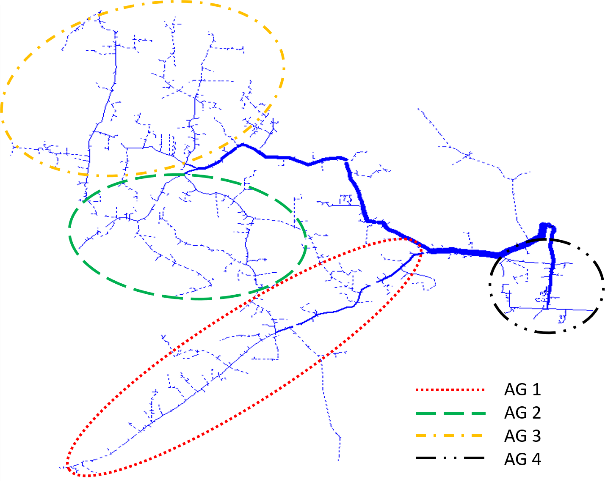
**Manual**

# **Basics**

1. The folder "11000\_node\_system" contains the 11,000-node test feeder settings for OpenDSS. Most importantly,

* “MasterWang2.dss” is the main file that will be called from the OPF solver and the state estimation solver,
* “Lines.dss” is the modified line file,
* “TransformersWang.dss” sets regulators to 0 tap, and
* “Loadswenbo.dss” is the modified loads file.
* Default capacitor and regulator control can be disabled by “Set controlmode=Off”.

2. The folder "python" contains linearization parameters and clustering settings based on the figure below, as well as the codes that generate these settings.



3. The folder "Result\_11000" contains plotted results.

* device.PNG is the device information.
* Folder “OPF” contains OPF solver results.
* Folder “SE” contains SE data and results.

4. The folder "spyder" contains all the codes for the proposed controllers. All of the intermediate and final results are retained in this folder. Specifically, the following files can be used to generate results directly:

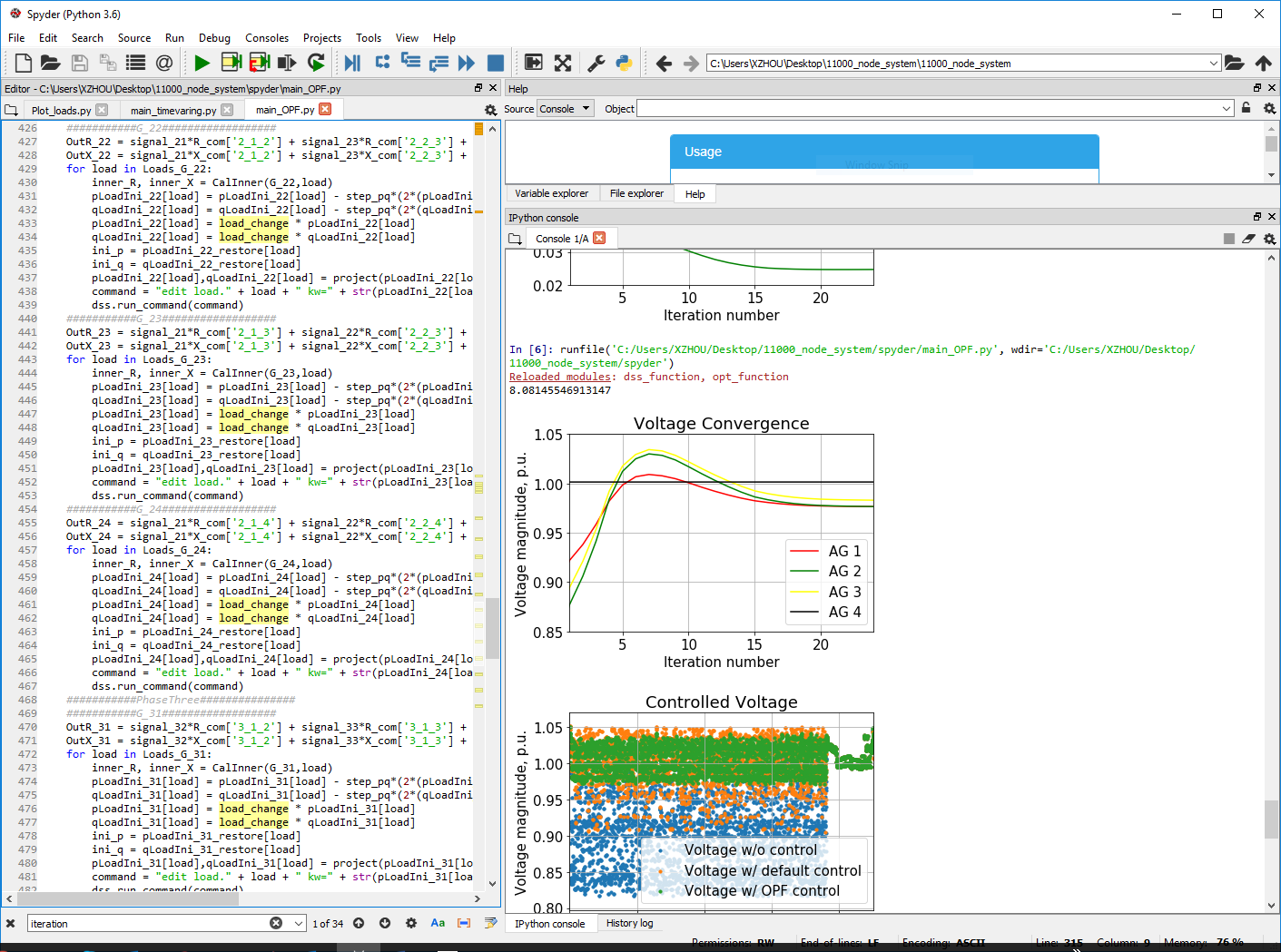
* main\_OPF.py : This file runs the OPF solver and plot line loss and voltages. The parameters that can be tuned include (1) stepsize for dual update, (2) stepsize for primal update, (3) regularization parameter for line loss, and (4) iteration numbers, from line 98-101.
* main\_SE.py : This file runs the state estimation and plots the results. The parameters that can be tuned include (1) stepsize for variable update, in line 706, (2) measurement penetration (0~1), which specifies the percentage of nodes with measurement capability, in line 645, (3) measurement error in line 646, and (4) iteration number in line 260.

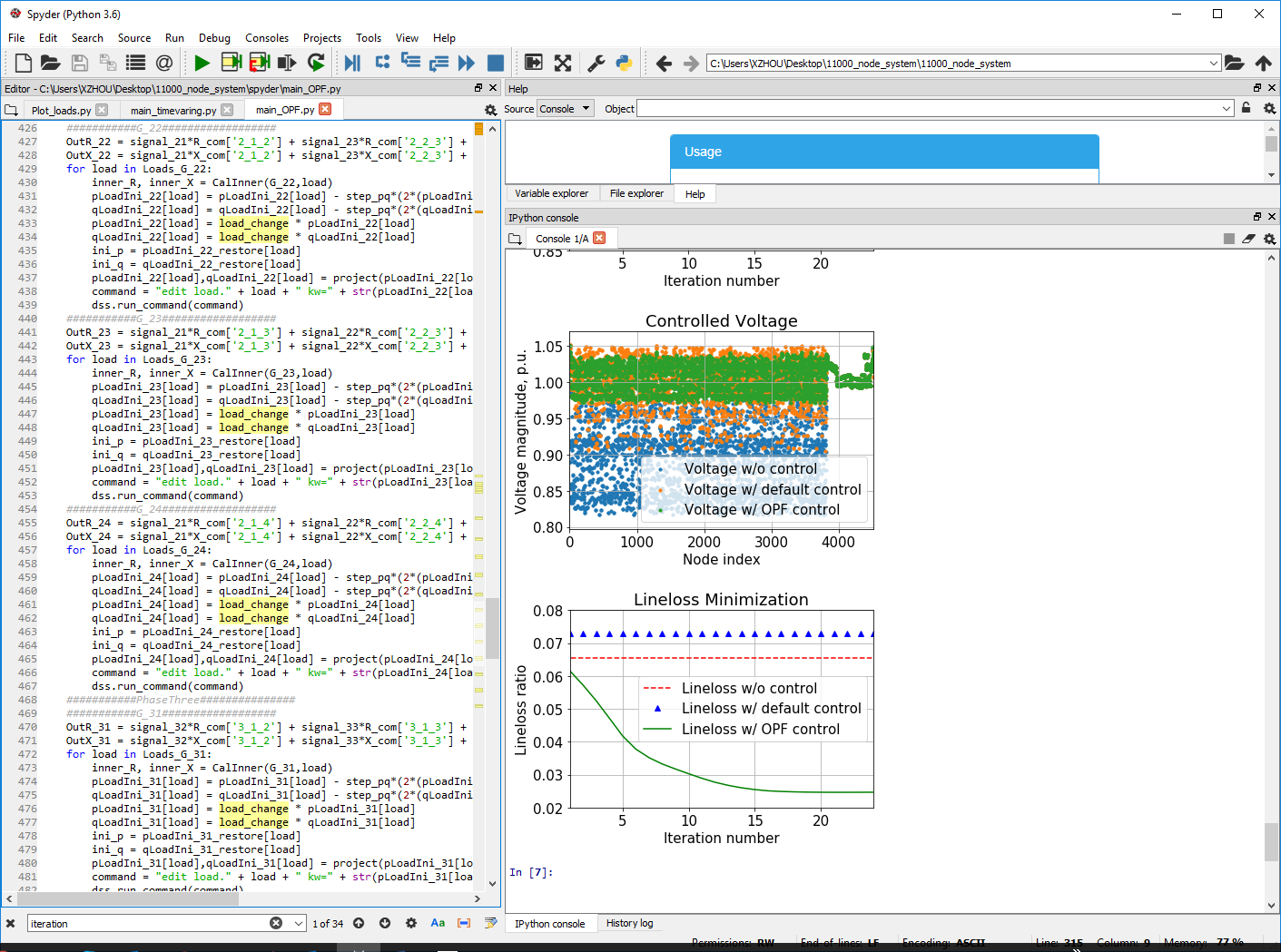
# **Example**

We provide an example for OPF solver and State Estimation solver in this part.

## OPF Solver

* Open main\_OPF.py
* Change file path to local address in lines 99, and 108—112
* Set iteration number in line 264
* Set constant stepsize for dual and primal variables in lines 295 and 296 respectively
* Set parameter of line loss in line 307
* (For the case with time-varying loads, set iteration number to larger than 250, and set load multipliers from Line 320 to Line 329.)
* Run the code and the following results will be displayed:
  + Simulation time
  + Convergence plot for voltages
  + Voltages over all the network nodes
  + Line loss plot





## State Estimation Solver

* Open main\_SE.py
* Change file path to local address in lines 99, and 108—112
* Set the ratio of nodes with measurement from 0 to 1 in Line 645, and accordingly, “measureLoads = measureSet(ratio)” in Line 647 will randomly pick nodes with voltage measurement.
* Set measurement error (standard deviation in Gaussian distribution for noise generation) in Line 646. Default value is 0.01.
* Set stepsize in Line 706. With more nodes that have measurement capability, the stepsize needs to be tuned down accordingly.
* Set in Line 707 whether the nodes have only voltage measurement capability or they can also measure power injections.
* Run the code and the following results will be displayed:
  + Simulation time
  + Convergence plot for estimated voltages
  + Estimated voltages over all the network nodes

