Brief introduction to the two programs

Rui Bo

1. Overview

Two generic programs, Continuation Power Flow (CPF) and State Estimation (SE), are implemented in MATLAB language. Book 'Computational Methods for Electric Power Systems' authored by Mariesa Crow is referenced during implementation of the programs, and the notations are in accordance with those in the book.

The file structure is as follows.

1) Files used for CPF program. There are 5 source code files and 1 test file.
   cpf.m: CPF solver
   cpf_predict.m: prediction step of CPF
   cpf_correctLambda.m: correct lambda in correction step
   cpf_correctVoltage.m: correct voltage in correction step
   drawPVcurves.m: draw PV curves
   test_cpf.m: test CPF solver
2) Files used for State Estimation program. There are 3 source code files and 1 test file.
   run_se.m: run state estimation
   doSE.m: perform state estimation
   outputsesoln.m: show state estimation results on screen
   test_se.m: test state estimation solver
3) Two case data files.
   case3bus_P6_6.m: a 3 bus system in problem 6.6 in book 'Computational Methods for Electric Power Systems' authored by Mariesa Crow
   case6bus.m: a 6 bus system in problem 3.6 in book 'Computational Methods for Electric Power Systems' authored by Mariesa Crow
4) In subfolder 'auxl', some auxiliary files are included, most of which are simplified from the corresponding MATPOWER v3.0.0 source code files. **So far, all the source codes keep highly compliance with MATPOWER package**. In addition, all the auxiliary files could be replaced with corresponding MATPOWER files with little change to the codes of these two programs.

Features of the two programs:

- ✓ All the source codes keep highly compliance with MATPOWER package and therefore could be incorporated into MATPOWER package with ease
- ✓ Source codes are self-explanatory
- ✓ Sufficient comments are provided in the source codes

NOTE: Execution of the two programs requires the MATPOWER package v3.0.0 is included in the searching path of MATLAB environment.

2. Continuation Power Flow (CPF) program

This program implements the continuation power flow solver and plots the PV curve as well as the prediction-correction trajectory.

2.1. Features

- ✓ Obtain the PV curve through simulation employing prediction-correction (predictor/corrector) method

- ✓ The PV curve as well as the prediction-correction trajectory could be plotted
- ✓ PV curve of every bus is available for plotting
- ✓ Load at (or near) the nose point of the PV curve, i.e., maximum load value, is obtained
- ✓ Simulation step-sizes for voltage predictor/corrector and lambda predictor/corrector could be specified by users to seek balance between running time and accuracy
- ✓ Tested on a given 6 bus system, and the 30 bus system included in MATPOWER

2.2. Limitations

- ✓ Sparse matrix techniques have not been employed
- ✓ Presently CPF with respect to load at only one bus is supported. It could be easily extended to load change at multiple buses
- ✓ Presently no more than one generator at each bus is assumed. It could be extended to support multiple generators at one bus with moderate coding
- ✓ Screen printouts are not very nicely formatted just for simplicity
- ✓ Enforcement of generator Q limit is not considered. It could be implemented with 'ENFORCE_Q_LIMS' option in MATPOWER power flow solver.
- ✓ It is not fully tested for the case with inconsecutive bus numbering

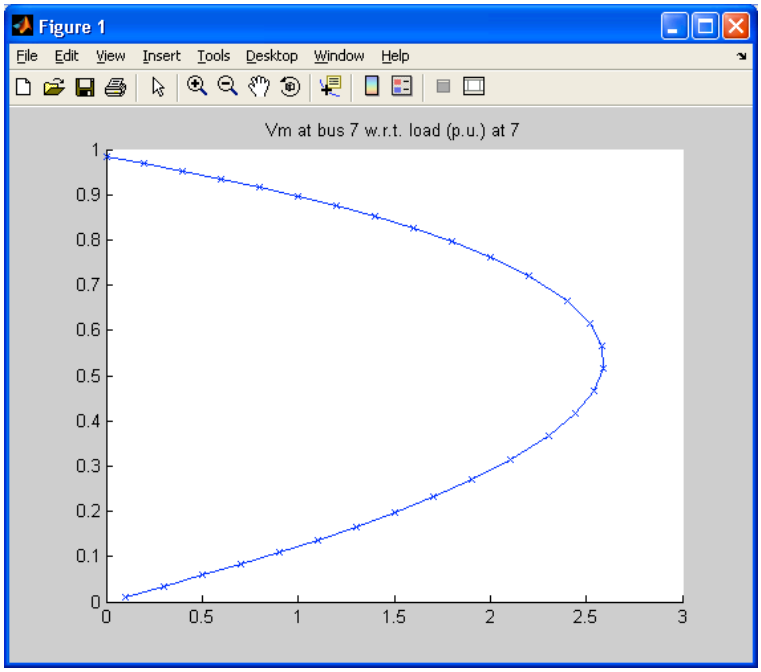
2.3. Demonstration

Running 'test_cpf.m' will generate the following outputs. The test case is a 30 bus case included in MATPOWER package. The PV curve with respect to load at bus 7 is plotted by the CPF program as follows.

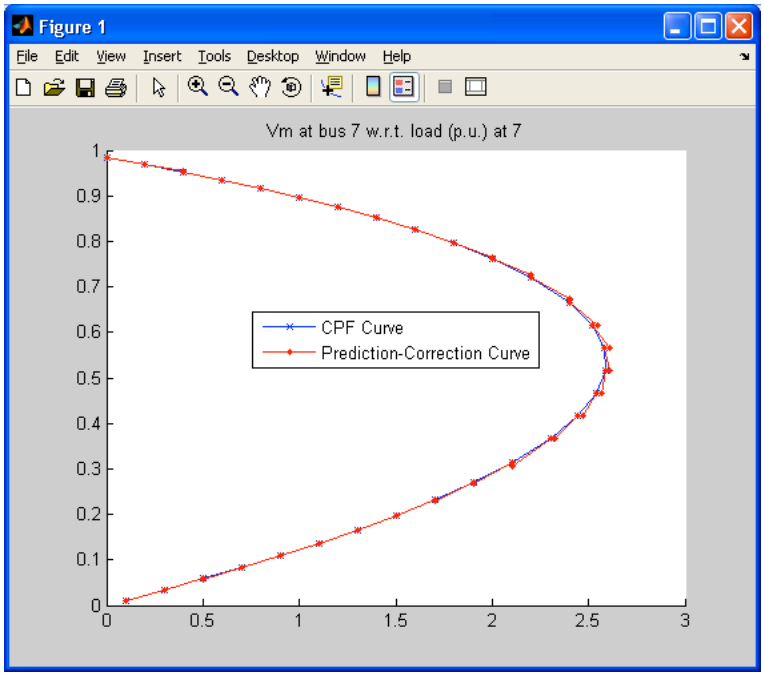
```
>> test_cpf

------------testing continuation power flow (CPF) solver
Start Phase 1: voltage prediction-correction (lambda increasing).
    [Info]: Approaching nose area of PV curve, or voltage correction fails.
    [Info]: 13 data points contained in phase 1.
Switch to Phase 2: lambda prediction-correction (voltage decreasing).
    [Info]: Leaving nose area of PV curve, or lambda correction fails.
    [Info]: 6 data points contained in phase 2.
Switch to Phase 3: voltage prediction-correction (lambda decreasing).
    [Info]: lambda is less than 0.
            CPF finished.
    [Info]: 11 data points contained in phase 3.
maximum lambda is 2.585415

Start plotting CPF curve(s)...
Plotting is done.
>>
```



The PV curve as well as the prediction-correction trajectory are plotted by the CPF program as follows.



3. State Estimation (SE) program

This program intends to improve the state estimation code included in MATPOWER v3.0.0 (in sub directory 'matpower3.0.0\extras\state_estimator') by defining a generic interface and rewrite some of the codes.

3.1. Features

✓ A generic interface of the state estimation is defined

- ✓ 8 types of measurements are supported including real power line flow through from end, real power line flow through to end, generator real power output, voltage angle, reactive power line flow through from end, reactive power line flow through to end, generator reactive power output, voltage magnitude (shortened for PF, PT, PG, Va, QF, QT, QG, Vm respectively)
- ✓ Measurement variances for each of the 8 kinds of measurements could be separately specified.

3.2. Limitations

- ✓ Presently bad data detection is not implemented
- ✓ The measurement indices of the interface may be further improved to facilitate the use of the program

3.3. Demonstration

Running 'test_se.m' will generate the following outputs. The test case is a 6 bus system, which is included in the state estimation program package.

```
================================================================================
|     Comparison of measurements and their estimations                         |
|     NOTE: In the order of PF, PT, PG, Va, QF, QT, QG, Vm (if applicable)     |
================================================================================
    Type         Index      Measurement   Estimation
                  (#)         (pu)          (pu)
 -----------  -----------  -----------   ----------
      PF          1          0.1200         0.1474
      PF          2          0.1000         0.0783
      PT          3         -0.0400        -0.0399
      PG          1          0.5800         0.5757
      PG          2          0.3000         0.3034
      PG          3          0.1400         0.1336
      Vm          2          1.0400         1.0258
      Vm          3          0.9800         0.9790

[Weighted sum of error squares]:    5.417915
```