

# EE 521 Power System Analysis and EE 523 Power System Stability and Control Algorithms

## Preamble and Control Inputs

```
tic;
addpath functions\
systemName = "ieee11"
powerFlowMethod = "NRPF"
useSparseDSA = false;
includeOPFScenarios = false;
showOPFFormulae = false;
showOPFValues = false;
numIterations = 50; %I don't wait for the system to converge,
printPowerFlowConvergenceMessages = false;
% neither do I care if the system converges earlier.
toleranceLimit = 1e-3; %mean of absolute values of
% corrections should be less than this for convergence to be achieved.
displayRawData = true;
displayYBus = true;
displayTables = true; %show busData, branchData, ybus,
% basically data structures which are not the final output.
printJacobians = true ; %Print Jacobians during NRPF iterations? Does not work if displayTables
printMismatches = false; %Print Mismatches during NRPF iterations? Does not work if displayTables
printCorrections = false;
disableTaps = false; %Disable Tap-changers when computing YBus?
showPlots = true;
displayResults = true;
reducedBranchColumnsCDFReading = true;
showImages = true; %might add iteration specific images later.
verboseCDFReading = false; %Will give a verbose output when reading CDF files.
MVAb = 100; %Currently the same for all systems in database.
```

```
folder_rawData = "rawData/"; %location of CDF .txt file for the system
file_rawData = strcat(folder_rawData, systemName, "cdf.txt"); %Exact location of CDF .txt file
folder_processedData = "processedData/";
% Should configure it to be read from the CDF file later.
latex_interpreter %for LaTeX typesetting in plots
```

## Read CDF file and store the data in neat MATLAB tables: busData and branchData.

```
[busData, branchData, N, numBranch] = ...
    readCDF(file_rawData, reducedBranchColumnsCDFReading, verboseCDFReading);
if displayTables && displayRawData
    displayRawDataAsTables(busData, branchData, N, numBranch);
end
```

**Extract  $Y_{Bus}$ , Adjacency List  $E$  from the branchData table.**

```
if useSparseDSA
    [nnzYBus, NYBus] = makeSparseYBus(busData, branchData, displayTables, displayYBus); %#ok
else
    [ybus, BMatrix, ~, ~, ~, E] = ybusGenerator(busData, branchData);
    ybusTable = array2table(ybus, VariableNames=[string(1:N)], RowNames=[string(1:N)]);

end

if ~useSparseDSA && displayTables && displayYBus
    display(ybusTable);
end
```

**Run Newton Raphson Power Flow and obtain a steady state snapshot of the system variables  $P_i, Q_i, V_i, \delta_i \forall$  buses  $i \in [1, N], i \in \mathbb{N}$**

```
[PSpecified, QSpecified, V, delta, ...
    listOfPQBuses, listOfPVBuses, nPQ, nPV, ...
    listOfNonSlackBuses] = initializeVectors(busData, N, MVAb);

if useSparseDSA
    doTheSparseThing(PSpecified, QSpecified, V, delta, nnzYBus, NYBus, busData); %#ok
else
    if contains(systemName, 'caseThree')
        resultsFromCaseTwo = load("processedData\ieee11-caseTwoResults");
        resultsFromCaseTwo = resultsFromCaseTwo.resultTable;
        wiggle = 0.35; %minimum 0.35 value for good result
        V = V*(1-wiggle) + wiggle*resultsFromCaseTwo.V;
        delta = delta*(1-wiggle) + wiggle*resultsFromCaseTwo.delta;
    end
    [P, Q, V, delta] = solveForPowerFlow(PSpecified, QSpecified, V, delta, ybus, BMatrix, E, N);

    resultTable = displayPowerFlowResults(N, P, Q, V, delta, displayResults);
    if contains(systemName, 'caseTwo')
        save("processedData\ieee11-caseTwoResults", "resultTable");
    end
end
```

**Compare obtained snapshot values of  $V_i$  and  $\delta_i$  against the ones given in the CDF file.**

```
plotPowerFlowResults(showPlots, useSparseDSA, V, busData, systemName, powerFlowMethod, delta);
```

**Economic Dispatch and Optimal Power Flow Calculations:**

```
if includeOPFScenarios && strcmp(systemName, 'ieee14') && ~useSparseDSA
    runOPFScenarios(busData, P, Q, V, delta, N, ybus, BMatrix, E, nPQ, nPV, listOfPQBuses, listOfPVBuses, listOfNonSlackBuses);
end
```

```
toc;
```

## Have a nice day!

In case you encounter a Java Heap Memory error, delete the above gif, or go to Preferences -> General -> Java Heap Memory and increase the allocated size.