# EE3123 Project

Umirbayev Alikhan

SID: 55810104

## Task 1

### 1

The analysis in normal steady-state operation is called a power flow analysis. The purpose of the analysis is finding the voltages (magnitudes and phases), currents, real and reactive power flows in a power system under specified generation and load conditions. In addition, line flows and losses can be calculated.

First of all, we usually have a minimal set of variables: V, δ, P, Q (voltage; delta-angle; active power; reactive power). All other variables can be obtained from this minimal set. Moreover, if we have any two variables of each bus (2 out of V, δ, P, Q), we can find the other two unknown variables.

Regarding the unknown and specified variables, we distinguish three types of buses: load bus, generator bus and slack bus.

Load bus (PQ bus) – Buses not having a generator

- Real and reactive powers (P and Q) are specified
- Bus voltage magnitude and phase angle (V and δ) will be calculated
- Power supplied to the power system is positive
- Power consumed from the system is negative.

Generator bus (PV bus)

- Voltage and real power supplied are specified
- Bus phase angle will be calculated during iteration
- Reactive power will be calculated after the case's solution is found

Slack bus (swing bus) – Special generator bus serving as the reference bus for the power system.

- Voltage is fixed – both magnitude and phase.
- The bus supplies whatever real or reactive power is necessary to balance the power flow in the system.

Before Power Flow Analysis, we make several assumptions:

- Generation supply = load demand + system losses.
- Voltage magnitudes of buses remain close to rated values.

- Generators operate within specified real and reactive power limits.
- Transformers and transmission lines are not overloaded.

Here is a table of type of buses in power flow analysis problems:

| Bus type | Voltage ($\|V_i\|\angle\delta_i$) | | Real power | | | Reactive power | | |
|---|---|---|---|---|---|---|---|---|
| | **Magnitude** | **Angle** | **Generation** | **Load** | **Net ($P_i$)** | **Generation** | **Load** | **Net ($Q_i$)** |
| Slack/Swing | Specified | Specified | Unknown | Specified | Unknown | Unknown | Specified | Unknown |
| Generator/PV | Specified | Unknown | Specified | Specified | Specified | Unknown | Specified | Unknown |
| Load/PQ | Unknown | Unknown | Specified | Specified | Specified | Specified | Specified | Specified |

First, we need to calculate the admittance matrix by using nodal analysis.

$I_k = V_k(Y_k + \sum_{m\neq k} Y_{km}) - \sum_{m\neq k} V_m Y_{mk}$

where $Y_k + \sum_{m\neq k} Y_{km}$ is admittance connected to bus k;

$Y_{mk}$ is Admittance between bus k and bus m.

$$\begin{bmatrix} I_1 \\ \dots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & \cdots & -Y_{1n} \\ \vdots & \ddots & \vdots \\ -Y_{n1} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ \dots \\ V_n \end{bmatrix}$$

$Y_{kk}$ = sum of all admittances connecting to node k

$Y_{kn}$ = $Y_{nk}$ = sum of all admittances connecting node k and node n

So, the system matrix can be formed: $I = Y_{bus}V$, where $Y_{bus}$ is the bus admittance matrix of order n x n; V is the bus voltage vector; I is the source current vector.

Bus voltages can also be solved as $V = Y_{bus}^{-1} I$

The bus voltages and currents can be used to calculate power flow.

$$\mathrm{P} = \mathrm{VI}\cos(\theta_1 - \theta_2)$$
$$\mathrm{Q} = \mathrm{VI}\sin(\theta_1 - \theta_2)$$

$$S = P + jQ$$
$$S = |V|.|I|\angle(\theta_1 - \theta_2) = V\, I^*$$
$S = V\, I^*$ so

$$S_k = V_k\, {I_k}^* = P_k + jQ_k = V_k \sum_{m=1} Y_{mk}^* V_m^*$$

$Y_{mk} = G_{mk} + jB_{mk}$, where $G_{mk}$ and $B_{mk}$ are the real and imaginary parts of the admittance matrix element $Y_{mk}$

Power flow equations

$$S_k = \sum_{m=1}^{n} V_k V_m \angle(\theta_k - \theta_m)(G_{mk} - jB_{mk})$$

$$= \sum_{m=1}^{n} V_k V_m (cos(\theta_k - \theta_m) + jsin((\theta_k - \theta_m))(G_{mk} - jB_{mk})$$

$$P_k = \sum_{m=1}^{n} V_k V_m (G_{mk} cos(\theta_k - \theta_m) + B_{mk} \sin((\theta_k - \theta_m))$$

$$Q_k = \sum_{m=1}^{n} V_k V_m (G_{mk} sin(\theta_k - \theta_m) - B_{mk} cos((\theta_k - \theta_m))$$

For each bus, if we know 2 variables, we have only 2 unknowns. Therefore, for 2n equations, we have 2n unknowns to solve.

However, if the equations are nonlinear, containing products and sine/cosine of unknowns. As a result, we cannot solve them using algebraic method, so we need numerical methods. The popular methods for power flow analysis are Gauss-Seidel method and Newton-Raphson method.

Or another way to solve is to simplify to DC power flow equation.

For DC approach we made additional assumptions:

- The resistance in all connecting lines is much less than the reactance on the transmission lines, i.e., $G_{km} \approx 0$, so we have:

  $$P_k = \sum_{m=1}^{n} V_k V_m (B_{mk} \sin((\theta_k - \theta_m))$$

  $$Q_k = \sum_{m=1}^{n} V_k V_m (-B_{mk} cos((\theta_k - \theta_m))$$

- The angle difference between bus voltages is very small.

  $$P_k = \sum_{m=1}^{n} V_k V_m (B_{mk} (\theta_k - \theta_m))$$

  $Q_k = \sum_{m=1}^{n} V_k V_m (-B_{mk})$ In $Q_k$, $V_k(V_k – V_m)$ is retained, so $Q_k \approx 0$

- Bus voltage is nearly 1 pu, i.e., voltages are similar in magnitude. Focus on real power because P >> Q.

  $$P_k = \sum_{m=1, m \neq k}^{n} (B_{mk} (\theta_k - \theta_m))$$

The final equation:

$$P_k = \sum_{m=1, m\neq k}^{n} (B_{mk}(\theta_k - \theta_m))$$

It is linear equations, so they are solvable without numerical methods.

We use the equation for each bus. The general equation can be written like:

$$[P] = [B][\theta]$$

[P]: n-dim vector of bus active power injections for buses 1, ..., n

[B]: n x n admittance matrix with R = 0

[θ]: n-dim vector of bus voltage angles for buses 1, ..., n

And therefore, power flows through branches are represented as:

$$[P_{br}] = ([b] \times [A])[\theta]$$

[$P_{br}$]: n-dim vector of branch flows (M is the number of branches)

[b]: M x M matrix ($b_{kk}$ is equal to the susceptance of line k and non-diagonal elements are zero)

[A]: M x n bus-branch incidence matrix

## 2

The main idea of all three methods – to solve the power flow equations. The Newton-Raphson method and the Gauss-Seidel method are numerical methods to solve AC power flow equations.

DC power flow – is the simplified approach to the power flow, where we linearize power flow equations, so it can be solved with algebraic methods without using numerical methods. The derivation of the DC power flow is described in first part of Task 1.

Advantages:

- Solvable without numerical methods
- Linear equations
- Simple

Disadvantages:

- Less accurate, especially for large systems

The Gauss-Seidel method is the simplest numerical method that is usually used for power flow problems. In this method, each component of current iteration depends on results from previous iterations. In case of power flow analysis, it is voltages value. For load buses, both angle and magnitude to be calculated, for generator buses, angles must be computed. Gauss-Seidel technique follows the iterative steps to reach the solution for the function f(x)=0:

- x=g(x)

- The initial value of function g(x[0]) using the initial guess x[0] that we need to make. The value of initial guess is usually based on the given data. However, the initial guessed value is usually Vi=1∠0 because of the first bus is usually slack bus.
- The next iteration value x[k+1]=g(x[k]).
- We need to continue the iterations until |x[k+1]−x[k]||≤E. E is the acceptable value and it represents the accuracy. The smaller value, the more accurate the method will be. However, it means that more iterations must be performed.
  The equation that is used to calculate the iterative value of voltage is:
  $$V_i^{[k+1]} = \frac{1}{Y_{ii}}\left(\frac{P_i^{net} - jQ_i^{net}}{V_i^{*[k]}} - \sum_{j\neq i} Y_{ij}V_j^{[k\ or\ k+1]}\right)$$
  $V_i$ is the voltage at bus I; $Y_{ij}$ is the Y-bus element; $P_i^{net}$ and $Q_i^{net}$ are net scheduled injected real power and reactive power respectively.

Advantages:

- It is the simplest method
- Small memory size required
- Each iteration is fast (low computational time per iteration)

Disadvantages:

- Not suitable for large systems. (Number of iterations is increasing directly with the number of buses)
- Slow rate of convergence
- Can miss the solution on large systems
- Not suitable for negative branch reactance

The Newton-Raphson method is the method of successive approximation. It is based on Taylor's expansion series:

$$f(x[0] + \Delta x[0]) = c$$
$$= f(x[0]) + f'(x[0])\Delta x[0] + \frac{1}{2!}f''(x[0])\Delta x[0]^2 + \frac{1}{3!}f'''(x[0])\Delta x[0]^3 + \cdots$$

If we neglect the higher order components since the value becomes too small, then we can use the first two components. So, we can obtain $\Delta x[0] = c - \frac{f(x[0])}{f\prime(x[0])} = \frac{\Delta f(x[0])}{f\prime(x[0])}$

As a result, $x[k+1] = x[k] + \Delta x[k]$. Or in general form, when c =0: $x[k+1] = x[k] - \frac{f(x[k])}{f\prime(x[k])}$

We continue iterations until |Δf[k]||≤E, where E is tolerance limit.

For power flow case we have the following:

$$x[k] = \begin{bmatrix} \delta i[k] \\ |Vi|[k] \end{bmatrix}, c = \begin{bmatrix} P_i^{net} \\ Q_i^{net} \end{bmatrix}, f = \begin{bmatrix} P_i^{[k]} \\ Q_i^{[k]} \end{bmatrix}, \qquad \begin{bmatrix} P_i^{net} \\ Q_i^{net} \end{bmatrix} - \begin{bmatrix} P_i^{[k]} \\ Q_i^{[k]} \end{bmatrix} = \begin{bmatrix} \Delta P_i^{[k]} \\ \Delta Q_i^{[k]} \end{bmatrix}$$

We introduce here the Jacobian matrix - partial derivative matrix

$$J = \begin{bmatrix} \frac{\partial P_i}{\partial \delta_i} & \frac{\partial P_i}{\partial |Vi|} \\ \frac{\partial Q_i}{\partial \delta_i} & \frac{\partial Q_i}{\partial |Vi|} \end{bmatrix}$$

Therefore: $\begin{bmatrix} {P_i}^{net} \\ {Q_i}^{net} \end{bmatrix} - \begin{bmatrix} {P_i}^{[k]} \\ {Q_i}^{[k]} \end{bmatrix} = \begin{bmatrix} \Delta {P_i}^{[k]} \\ \Delta {Q_i}^{[k]} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_i}{\partial \delta_i}[k] & \frac{\partial P_i}{\partial |Vi|}[k] \\ \frac{\partial Q_i}{\partial \delta_i}[k] & \frac{\partial Q_i}{\partial |Vi|}[k] \end{bmatrix} \begin{bmatrix} \Delta \delta i[k] \\ \Delta |Vi|[k] \end{bmatrix}$

$$\begin{bmatrix} \Delta \delta i[k] \\ \Delta |Vi|[k] \end{bmatrix} = \begin{bmatrix} \frac{\partial P_i}{\partial \delta_i}[k] & \frac{\partial P_i}{\partial |Vi|}[k] \\ \frac{\partial Q_i}{\partial \delta_i}[k] & \frac{\partial Q_i}{\partial |Vi|}[k] \end{bmatrix}^{-1} \begin{bmatrix} \Delta {P_i}^{[k]} \\ \Delta {Q_i}^{[k]} \end{bmatrix}$$

So, the new values are calculated by:

$$\begin{bmatrix} \delta i[k] \\ |Vi|[k] \end{bmatrix} = \begin{bmatrix} \delta i[k-1] \\ |Vi|[k-1] \end{bmatrix} + \begin{bmatrix} \Delta \delta i[k] \\ \Delta |Vi|[k] \end{bmatrix}$$

Continue until $\begin{bmatrix} \Delta {P_i}^{[k]} \\ \Delta {Q_i}^{[k]} \end{bmatrix} \leq E$

Advantages:

- More accurate, faster, reliable
- Suitable for large systems
- Fast convergence

Disadvantages:

- Larger computation time per iteration
- More complicated
- More memory required

In summary, Power flow equations are nonlinear, so we need numerical techniques such as the Gauss-Seidel, the Newton-Raphson, DC power flow methods to solve these equations. In general, the Gauss-Seidel method is the simplest but converges slower and with lower accuracy than the Newton-Raphson technique. However, each iteration in the Newton-Raphson method is computed longer than in the Gauss-Seidel method because you need to calculate the Jacobian matrix every time. In DC power flow method, the voltage is assumed constant at all buses and the problem becomes linear, so you can easily solve equations, but the accuracy may be sacrificed.

# Task 2

## 1

This task requires to use the Newton-Raphson method to compute AC power equations for IEEE 14-bus and IEEE 118-bus power test cases. The input data, including the power test cases data and the bus admittance matrix, are provided. I tried to implement the display output format similar to MATPOWER runpf(). The procedure of the Newton-Raphson method is standard. First, I initialize the program. There is an input to choose between 14 and 118 test case data. Next, I load all provided data from the test cases including the bus admittance matrix. Then I start the Newton-Raphson method.

First, calculate the active and reactive power for the bus using the following equations:

$$P_k = \sum_{m=1}^{n} V_k V_m (G_{mk} cos(\theta_k - \theta_m) + B_{mk} \sin((\theta_k - \theta_m))$$

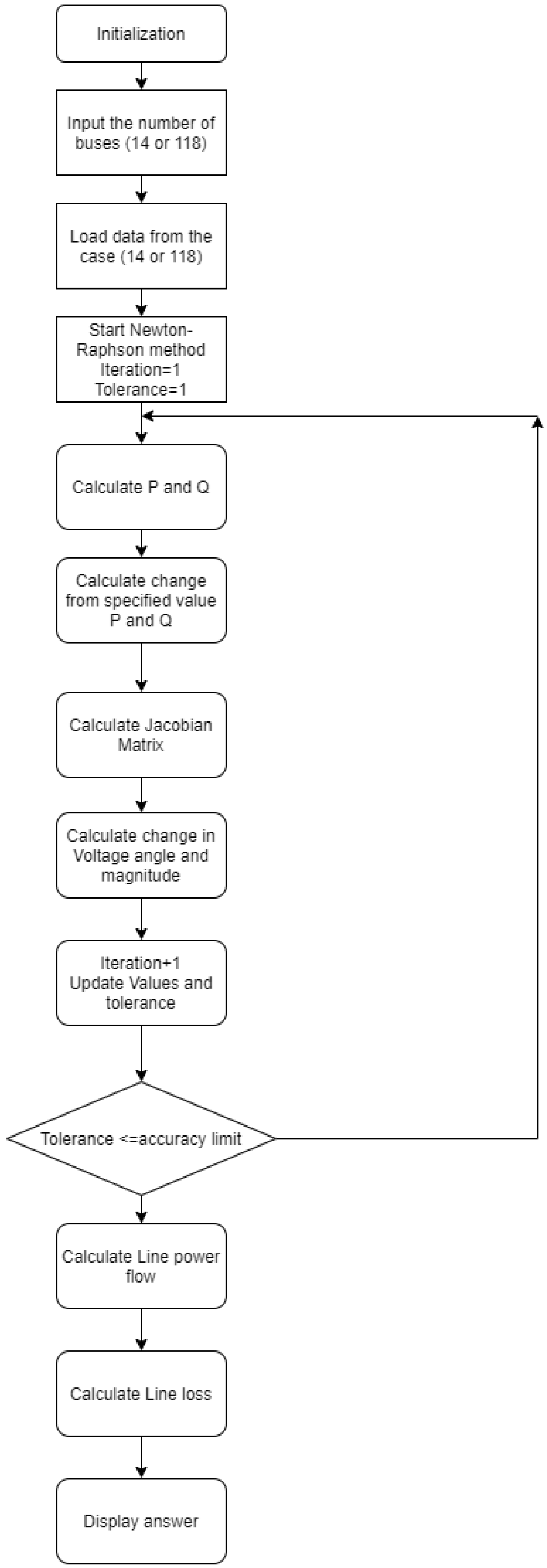
$$Q_k = \sum_{m=1}^{n} V_k V_m (G_{mk} sin(\theta_k - \theta_m) - B_{mk} cos((\theta_k - \theta_m))$$

Next, I compute the change from the specified values of P and Q derived from the difference of generation and load value from the input data. I form the mismatch vector according to this change.

Then, I start to compute the Jacobian matrix $J = \begin{bmatrix} \frac{\partial P_i}{\partial \delta_i} & \frac{\partial P_i}{\partial |Vi|} \\ \frac{\partial Q_i}{\partial \delta_i} & \frac{\partial Q_i}{\partial |Vi|} \end{bmatrix}$. At last, I update the iteration value of voltage angle and magnitude according to the Jacobian and mismatch vector (correction matrix). If the tolerance is smaller than the specified accuracy limit, the Newtom-Raphson method is finished.

After this, I calculate the line power flow and line loss using the complex power $S_k = V_k I_k{}^*$

At the end, the answer is displayed showing the injection, generation, load power at each bus and power flow for each line with loss.


Initialization
Input the number of buses (14 or 118)
Load data from the case (14 or 118)
Start Newton-Raphson method Iteration=1 Tolerance=1
Calculate P and Q
Calculate change from specified value P and Q
Calculate Jacobian Matrix
Calculate change in Voltage angle and magnitude
Iteration+1 Update Values and tolerance
Tolerance <=accuracy limit
Calculate Line power flow
Calculate Line loss
Display answer

## 2

DC is the simplified approach to power flow equations. This task requires to use the DC method to perform power flow analysis for IEEE 14-bus and IEEE 118-bus power test cases. The input data, including the power test cases data and the bus admittance matrix, are provided. I tried to implement the display output format similar to MATPOWER runpf(). Assumptions for DC method were made.

I assumed that the bus 1 is the slack bus, so the first row of $\boldsymbol{P}$ and the first row and column of $\boldsymbol{B}$ are disregarded, the voltage angle of the slack bus is set to zero.

The active power injection at each bus can be computed as the difference between the generation and the load:

$$\mathbf{P} = \mathbf{P}_{\boldsymbol{G}} - \mathbf{P}_{\boldsymbol{L}}$$

Voltage phase can be computed as follows:

$$\boldsymbol{\theta} = \boldsymbol{B}^{-1}\boldsymbol{P}$$

**B** is the nxn admittance matrix with R = 0

**P** is the nx1 vector of bus active power injections for buses 1, …, n

$\boldsymbol{\theta}$ is nx1 vector of bus voltage angles for buses 1, …, n

The iterative value of active power injection at bus i is computed as:

$$P_i = \sum_{\substack{j=1, \\ j \neq k}}^{N} B_{ij}\left(\theta_i - \theta_j\right)$$

So the power flow between bus k and j is computed as follows:

$$P_{kj} = B_{kj}\left(\theta_k - \theta_j\right)$$

To compare results of DC and Newton-Raphson methods, there is a difference in obtained values of angle. But the most distinguishable feature of DC approach is line power flow. In the Newton-Raphson Ac method, I got slightly different values between active power values for from bus and to bus directions, so there is power loss at the line. However, in DC method, the magnitude of active power for both directions are the same, so there is no line loss at all!

In addition, for the case118, there are larger difference between values resulted from DC and Newton-Raphson method, so I assume that DC may result in larger error in larger systems.

# Task 3

The task requires to perform cascading failure process and identify the failure propagation path showing which power node/link fails at each time step. The output is the history of failed lines and the time step when the line is failed. I did not include the initial failure set into the output. In addition, there is a sorted list of all failed lines including the initial failure set.

Here are the results:

1) failure_set = [93]:
   The history of affected lines and time excluding initial set
   |Line | Time|

| Line | Time |
|---|---|
| 100 | 2 |
| 95 | 3 |
| 94 | 4 |
| 89 | 5 |
| 88 | 6 |
| 102 | 7 |
| 101 | 8 |
| 30 | 9 |
| 54 | 10 |
| 45 | 11 |
| 44 | 12 |
| 60 | 13 |
| 58 | 14 |
| 57 | 15 |
| 122 | 16 |
| 103 | 17 |
| 119 | 18 |
| 118 | 19 |
| 123 | 20 |
| 115 | 21 |
| 112 | 22 |
| 104 | 23 |
| 150 | 24 |
| 141 | 25 |
| 146 | 26 |
| 145 | 27 |
| 151 | 28 |
| 92 | 29 |
| 65 | 30 |
| 171 | 31 |
| 64 | 32 |
| 148 | 33 |
| 152 | 34 |
| 149 | 35 |
| 175 | 36 |

2) failure_set = [93, 65];
   The history of affected lines and time excluding initial set
   |Line | Time|

| | |
|---|---|
| 100 | 2 |
| 95 | 3 |
| 94 | 4 |
| 89 | 5 |
| 88 | 6 |
| 102 | 7 |
| 101 | 8 |
| 30 | 9 |
| 54 | 10 |
| 45 | 11 |
| 44 | 12 |
| 60 | 13 |
| 58 | 14 |
| 57 | 15 |
| 122 | 16 |
| 103 | 17 |
| 119 | 18 |
| 118 | 19 |
| 123 | 20 |
| 115 | 21 |
| 112 | 22 |
| 104 | 23 |
| 150 | 24 |
| 141 | 25 |
| 146 | 26 |
| 145 | 27 |
| 151 | 28 |
| 92 | 29 |
| 171 | 30 |
| 64 | 31 |
| 148 | 32 |
| 152 | 33 |
| 149 | 34 |
| 175 | 35 |

3) failure_set = [23, 93, 102];
The history of affected lines and time excluding initial set

| Line | Time |
|---|---|
| 54 | 2 |
| 21 | 3 |
| 19 | 4 |
| 18 | 5 |
| 25 | 6 |
| 30 | 7 |
| 66 | 8 |
| 59 | 9 |
| 100 | 10 |
| 65 | 11 |
| 63 | 12 |
| 103 | 13 |
| 121 | 14 |
| 94 | 15 |
| 98 | 16 |
| 92 | 17 |
| 22 | 18 |
| 39 | 19 |
| 171 | 20 |

| | |
|---|---|
| 32 | 21 |
| 37 | 22 |
| 88 | 23 |
| 89 | 24 |
| 77 | 25 |
| 75 | 26 |
| 80 | 27 |
| 78 | 28 |
| 76 | 29 |
| 74 | 30 |
| 72 | 31 |
| 119 | 32 |
| 118 | 33 |
| 123 | 34 |
| 115 | 35 |
| 112 | 36 |
| 104 | 37 |
| 150 | 38 |
| 141 | 39 |
| 146 | 40 |
| 145 | 41 |
| 151 | 42 |
| 20 | 43 |
| 64 | 44 |
| 152 | 45 |
| 148 | 46 |
| 149 | 47 |
| 67 | 48 |
| 175 | 49 |

As a bonus, I have tried to implement the AC power flow:

1) failure_set = [93]
   The history of affected lines and time excluding initial set

   | Line | Time |
   |---|---|
   | 30 | 2 |
   | 66 | 3 |
   | 60 | 4 |
   | 100 | 5 |
   | 54 | 6 |
   | 45 | 7 |
   | 44 | 8 |
   | 1 | 9 |
   | 2 | 10 |
   | 171 | 11 |
   | 18 | 12 |
   | 175 | 13 |

2) failure_set = [93, 65];
   The history of affected lines and time excluding initial set

   | Line | Time |
   |---|---|
   | 30 | 2 |
   | 66 | 3 |
   | 60 | 4 |
   | 61 | 5 |
   | 100 | 6 |
   | 54 | 7 |
   | 45 | 8 |

| | |
|---|---|
| 44 | 9 |
| 64 | 10 |
| 1 | 11 |
| 2 | 12 |
| 102 | 13 |
| 67 | 14 |
| 179 | 15 |
| 151 | 16 |
| 80 | 17 |
| 171 | 18 |
| 18 | 19 |
| 175 | 20 |

3) failure_set = [23, 93, 102];
   The history of affected lines and time excluding initial set
   |Line | Time|

| | |
|---|---|
| 30 | 2 |
| 95 | 3 |
| 94 | 4 |
| 101 | 5 |
| 98 | 6 |
| 100 | 7 |
| 54 | 8 |
| 44 | 9 |
| 45 | 10 |
| 2 | 11 |
| 1 | 12 |
| 92 | 13 |
| 89 | 14 |
| 88 | 15 |
| 171 | 16 |
| 60 | 17 |
| 57 | 18 |
| 58 | 19 |
| 80 | 20 |
| 26 | 21 |
| 25 | 22 |
| 151 | 23 |
| 46 | 24 |
| 39 | 25 |
| 179 | 26 |
| 18 | 27 |
| 175 | 28 |

There are severe differences between AC and DC results. In my opinion, the cascade simulations shows that the DC model can lead to inaccurate cascade predictions. In our case, the large network (IEEE 118-bus), the DC model overestimates the cascade. The underlying problem of DC model inaccuracy is the assumptions and simplifications that are made for this approach (ignoring power losses, reactive power flows; assuming that voltages at all buses are equal; etc.). As a result, even small errors in each line flows models can accumulate during the cascade failure process and result in significant differences at the end of the cascade propagation.