\[
\begin{bmatrix}
\tilde{P}_2 - \tilde{P}_2(\tilde{V}) \\
\tilde{P}_3 - \tilde{P}_3(\tilde{V}) \\
\vdots \\
Q_n - Q_n(\tilde{V})
\end{bmatrix} = 
\begin{bmatrix}
\Delta P_2 \\
\Delta P_3 \\
\vdots \\
\Delta Q_n
\end{bmatrix} 
\begin{bmatrix}
\Delta \theta_2 \\
\Delta \theta_3 \\
\vdots \\
\Delta \theta_n
\end{bmatrix}.
\]

\[
J = 
\begin{bmatrix}
\frac{\partial \tilde{P}_2}{\partial \theta_2} & \frac{\partial \tilde{P}_2}{\partial \theta_3} & \cdots & \frac{\partial \tilde{P}_2}{\partial \theta_n} \\
\frac{\partial \tilde{P}_3}{\partial \theta_2} & \frac{\partial \tilde{P}_3}{\partial \theta_3} & \cdots & \frac{\partial \tilde{P}_3}{\partial \theta_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial Q_n}{\partial \theta_2} & \frac{\partial Q_n}{\partial \theta_3} & \cdots & \frac{\partial Q_n}{\partial \theta_n}
\end{bmatrix}
\]
\[ \overline{V} \overline{I} \Rightarrow \overline{S} = \overline{V} \overline{I}^* \]

\[ \overline{S} = \overline{V} (\overline{Y} \overline{I})^* \]

\[ S = V_1 \sum_{k=1}^{n} y_{k1}^* \cdot V_{i1}^* \]

\[ S_2 = \]

\[ S_n \]
OVERVIEW OF POWER FLOW

Power Flow Problems

General Information

1. Represent the transmission system in \( Y_{bus} \) matrix

2. \( Y_{bus} \) elements are based on admittances of transmission lines, transformers, and shunt elements on the transmission system

3. Treat generators and loads as injections of real and reactive power.

4. Normally know P and Q at load buses, and \(|\bar{V}|\) and P at generators.

5. You want to find \(|\bar{V}|\) and \(\angle V\) at each bus, from solution of set of complex, non-linear equations.

6. You can later compute all line flows of P and Q using bus voltages and angles.

7. \( \Sigma P_{gen} = \Sigma P_{Load} + \Sigma P_{losses} \), but losses aren’t known.

8. So we need a slack bus for simulation purposes, where we specify voltage magnitude and angle, and leave \(P_i, Q_i\) to be determined when all done.

9. Complex power flow equation:
   Number the slack bus as Bus 1

   \[
   \bar{S}_i = \bar{V}_i \sum_{k=1}^{n} y_{ik} V_i^* \quad \text{for} \quad i \in [2, n]
   \]

10. Rectangular form:
    \( y_{ij} = g_{ij} + jb_{ij} \), recall, equation uses \( y_{ij}^* \)

    \[
    P_i = \sum_{k=1}^{n} |V_i||V_k| [g_{ik} \cos(\theta_i - \theta_k) + b_{ik} \sin(\theta_i - \theta_k)] \quad \text{for} \quad i \in [2, n]
    \]

    \[
    Q_i = \sum_{k=1}^{n} |V_i||V_k| [g_{ik} \sin(\theta_i - \theta_k) - b_{ik} \cos(\theta_i - \theta_k)] \quad \text{for} \quad i \in [2, n]
    \]
Gauss Iteration

1. Put equations in the form:
   \[ \bar{V}_i = h_i(\bar{V}) = \frac{1}{y_{ii}} \left[ \frac{S_i^*}{V_i^*} - \sum_{k=1,k\neq i}^{n} \frac{y_{ik}}{V_k} \bar{V}_k \right] \text{ for } i \in [2, n] \]

2. Make initial guesses for each \(|V| \angle \theta\), generally, \(1 \angle 0^\circ\)

3. Solve the set of equations:
   \[
   \frac{V_2^{m+1}}{V_3^{m+1}} = h_2(\frac{V_2^m, V_3^m, \ldots, V_n^m}{V_2^m, V_3^m, \ldots, V_n^m})
   \]
   \[
   \frac{V_4^{m+1}}{V_3^{m+1}} = h_4(\frac{V_2^{m+1}, V_3^{m+1}, \ldots, V_n^{m+1}}{V_2^m, V_3^m, \ldots, V_n^m})
   \]
   \[
   \vdots
   \]
   \[
   \frac{V_n^{m+1}}{V_n^m} = h_n(\frac{V_2^{m+1}, V_3^{m+1}, \ldots, V_n^{m+1}}{V_2^m, V_3^m, \ldots, V_n^{m}})
   \]

4. Until \( \|\bar{V}\| \leq \epsilon \)

5. If you are given a \( P, |V| \) bus: Compute \( Q \) for that bus before compute \( \bar{V}_i \) for that step. Plug in the known \( |V_j| \)'s.

Gauss-Seidel Method

1. Basically the same as Gauss Iteration, but now you use the updated voltages as you find them.

2. Make initial guesses for each \(|V| \angle \theta\), generally, \(1 \angle 0^\circ\)

3. Solve the set of equations:
   \[
   \frac{V_2^{m+1}}{V_3^{m+1}} = h_2(\frac{V_2^m, V_3^m, \ldots, V_n^m}{V_2^m, V_3^m, \ldots, V_n^m})
   \]
   \[
   \frac{V_4^{m+1}}{V_3^{m+1}} = h_4(\frac{V_2^{m+1}, V_3^{m+1}, \ldots, V_n^{m+1}}{V_2^m, V_3^m, \ldots, V_n^{m}})
   \]
   \[
   \vdots
   \]
   \[
   \frac{V_n^{m+1}}{V_n^m} = h_n(\frac{V_2^{m+1}, V_3^{m+1}, \ldots, V_n^{m+1}}{V_2^m, V_3^m, \ldots, V_n^{m}})
   \]

4. Until \( \|\bar{V}\| \leq \epsilon \)
Newton-Raphson Method

1. Use the rectangular form of the power flow equations.

2. Define:

\[
x = \begin{bmatrix}
\theta_2 \\
\theta_3 \\
\vdots \\
\theta_n \\
\vdots \\
|V_2| \\
|V_n|
\end{bmatrix}
\]

3. Define mismatch vector as:

\[
\Delta h(x^n) = \begin{bmatrix}
P_2 - P_2(x^n) \\
P_3 - P_3(x^n) \\
\vdots \\
P_n - P_n(x^n) \\
\vdots \\
Q_2 - Q_2(x^n) \\
\vdots \\
Q_n - Q_n(x^n)
\end{bmatrix}
\]

4. Again, make initial guesses for all \(|V_i|, \theta_i\)

5. Compute Jacobian, and evaluate for the current \(x^n\)

6. Solve: \(J(x^n) \Delta x^n = \Delta h(x)^n\) for \(\Delta x\)

7. \(x^{n+1} = x^n + \Delta x\)

8. Recompute mismatch

9. Repeat until \(\|\Delta h(x)^{n+1}\|\) \(\leq \epsilon\)
10. Eliminate equation for Q at any P, |V| buses

11. When converged, go back and find P and Q at slack bus, and Q at all P, |V| buses

Fast Decoupled Load Flow

1. Approximation to speed up Newton-Raphson

2. Assume that $g_{ik}$ is small enough to neglect at each bus.

3. Assume that $\theta_i - \theta_j$ is small, such that:
   \[
   \sin(\theta_i - \theta_j) = 0 \text{ and } \\
   \cos(\theta_i - \theta_j) = 1
   \]

4. Then Jacobian is of the form:

   \[
   J(x) = \begin{bmatrix}
   [B] & 0 \\
   0 & [B]
   \end{bmatrix}
   \]

   where:

   \[
   B = \begin{bmatrix}
   b_{22} & b_{23} & \cdots & b_{2n} \\
   b_{32} & b_{33} & \cdots & b_{3n} \\
   \vdots & \vdots & \ddots & \vdots \\
   b_{n2} & b_{n3} & \cdots & b_{nn}
   \end{bmatrix}
   \]

5. Then solve for $\Delta \theta$ and $\Delta |V|$ in the following:

   \[
   -[B]\Delta \theta = \begin{bmatrix}
   \frac{\Delta P_1}{|V_2|} \\
   \frac{\Delta P_2}{|V_3|} \\
   \vdots \\
   \frac{\Delta P_n}{|V_n|}
   \end{bmatrix}
   \]
\[-[B]|\Delta V| = \begin{bmatrix}
\Delta Q_2 \\
|V_2|
\hline
\Delta Q_3 \\
|V_3|
\hline
\vdots
\hline
\Delta Q_n \\
|V_n|
\end{bmatrix}\]

6. Then \(|V|^{n+1} = |V|^n + \Delta|V|^n\) and \(\theta^{n+1} = \theta^n + \Delta \theta^n\)

7. Until within tolerate on mismatch vector:
   \(||\Delta P|| \leq \epsilon\) and \(||\Delta Q|| \leq \epsilon\)