NUMERICAL SOLUTION OF THE POWER FLOW

It is rare the case when it is possible to find an analytical solution to the power flow equations; thus, one needs to resort to numerical computation schemes. One such scheme is the Newton-Raphson (N-R) method (sometimes referred to as Newton’s method) which is actually the industry standard implemented in most commercial power flow solvers. Next, we review the basic ideas behind the N-R method.

Figure 16.1:

The Newton-Raphson Method

Consider the real-valued function $f(x)$ depicted in Fig. 16.1. We would like to find the values of $x$ for which $f(x) = 0$; in the specific case of the function $f(x)$ in Fig. 16.1, there exists two such values, $x^*$ and $x^\dagger$. We will now discuss an iterative algorithm known as Newton-Raphson method that solves the problem.
• Initially, we choose some estimate, $x^0$, for what the solution of $f(x) = 0$ would be. Unless we are very lucky, this value will be such that $f(x^0) \neq 0$.

• By using the Taylor series expansion of $f$ around $x^0$, we can write:

$$f(x^0 + \Delta x) \approx f(x^0) + \frac{df}{dx}igg|_{x^0} \Delta x,$$

(16.1)

for some small $\Delta x$. In practice, what (16.1) says is that for small $\Delta x$, i.e., in a small neighborhood around $x^0$, we can approximate the nonlinear function $f$ by a straight line tangent to $f(x)$ at $x^0$.

• Now, let $\Delta x^0$ denote the value of $\Delta x$ that would result in the straight line in (16.1) intersecting the horizontal axis, i.e.,

$$f(x^0) + \frac{df}{dx}igg|_{x^0} \Delta x^0 = 0,$$

from where it follows that

$$\Delta x^0 = -f(x^0) \left( \frac{df}{dx}igg|_{x^0} \right)^{-1}.$$

With the point of view that the straight line in (16.1) is an approximation of $f$, we can think of the point $x^1 := x^0 + \Delta x^0$ as an estimate of the solution to $f(x) = 0$; thus, we have

$$x^1 = x^0 - f(x^0) \left( \frac{df}{dx}igg|_{x^0} \right)^{-1}.$$

• We can use now $x^1$ instead of $x^0$ and repeat the procedure:

$$f(x^1 + \Delta x) \approx f(x^1) + \frac{df}{dx}igg|_{x^1} \Delta x,$$

which by equating to zero yields yields:

$$\Delta x^1 = -\left( \frac{df}{dx}igg|_{x^1} \right)^{-1} f(x^1),$$

and by defining $x^2 := x^1 + \Delta x^1$, it follows that:

$$x^2 = x^1 - \left( \frac{df}{dx}igg|_{x^1} \right)^{-1} f(x^1).$$

• The iteration scheme is now clear:

$$x^{\nu+1} = x^{\nu} - \left( \frac{df}{dx}igg|_{x^{\nu}} \right)^{-1} f(x^{\nu}), \quad \nu = 0, 1, 2, \ldots,$$

for some $x^0$ chosen as the initial guess.
• The algorithm terminates when \(|x^{\nu+1} - x^{\nu}| < \varepsilon\), for some \(\varepsilon > 0\) small.

The iterative procedure described above is graphically depicted in Fig. 16.1. Note that in the example in Fig. (16.1), the choice of \(x^0\) resulted in the algorithm converging to \(x^*\), i.e., the solution found depends on the initial guess.

Consider now the \(n\)-dimensional case, i.e., we would like to find the values of \(x_1, x_2, \ldots, x_n\) for which

\[
\begin{align*}
    f_1(x_1, x_2, \ldots, x_n) &= 0, \\
    f_2(x_1, x_2, \ldots, x_n) &= 0, \\
    &\vdots \\
    f_n(x_1, x_2, \ldots, x_n) &= 0,
\end{align*}
\]  

(16.2)

(16.3)

where \(f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, n\). The Newton-Raphson iteration can be constructed as follows:

• Initially, we choose some \(x_1^0, x_2^0, \ldots, x_n^0\) as estimates of the values of \(x_1, x_2, \ldots, x_n\) that would solve the set of equations in (16.3).

• The Taylor series expansion of \(f_i, i = 1, 2, \ldots, n\), around \(x^0 := [x_1^0, x_2^0, \ldots, x_n^0]^\top\) in this case is:

\[
f_i(x_1^0 + \Delta x_1, \ldots, x_n^0 + \Delta x_n) = f_i(x_1^0, \ldots, x_n^0) + \left. \frac{\partial f_i}{\partial x_1} \right|_{x^0} \Delta x_1 + \left. \frac{\partial f_i}{\partial x_2} \right|_{x^0} \Delta x_2 + \cdots + \left. \frac{\partial f_i}{\partial x_n} \right|_{x^0} \Delta x_n \\
\quad = f_i(x^0) + \begin{bmatrix}
  \left. \frac{\partial f_1}{\partial x_1} \right|_{x^0} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x^0} & \cdots & \left. \frac{\partial f_1}{\partial x_n} \right|_{x^0} \\
  \left. \frac{\partial f_2}{\partial x_1} \right|_{x^0} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x^0} & \cdots & \left. \frac{\partial f_2}{\partial x_n} \right|_{x^0} \\
  \vdots & \vdots & \ddots & \vdots \\
  \left. \frac{\partial f_n}{\partial x_1} \right|_{x^0} & \left. \frac{\partial f_n}{\partial x_2} \right|_{x^0} & \cdots & \left. \frac{\partial f_n}{\partial x_n} \right|_{x^0}
\end{bmatrix} \begin{bmatrix}
  \Delta x_1 \\
  \Delta x_2 \\
  \vdots \\
  \Delta x_n
\end{bmatrix}
\]

for some \(\Delta x_1, \Delta x_2, \ldots, \Delta x_n\) small.

• Define

\[
f(x) = \begin{bmatrix}
  f_1(x) \\
  f_2(x) \\
  \vdots \\
  f_n(x)
\end{bmatrix}, \quad (16.4)
\]
then we can write

\[
f(x^0 + \Delta x) \simeq f(x^0) + \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} \bigg|_{x^0} & \frac{\partial f_1}{\partial x_2} \bigg|_{x^0} & \cdots & \frac{\partial f_1}{\partial x_n} \bigg|_{x^0} \\
\frac{\partial f_2}{\partial x_1} \bigg|_{x^0} & \frac{\partial f_2}{\partial x_2} \bigg|_{x^0} & \cdots & \frac{\partial f_2}{\partial x_n} \bigg|_{x^0} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial x_1} \bigg|_{x^0} & \frac{\partial f_n}{\partial x_2} \bigg|_{x^0} & \cdots & \frac{\partial f_n}{\partial x_n} \bigg|_{x^0}
\end{bmatrix} \begin{bmatrix}
\Delta x_1 \\
\Delta x_2 \\
\vdots \\
\Delta x_n
\end{bmatrix}
\]

\[= J(x^0) \Delta x \] (Jacobian matrix)

which by equating to zero as in the one-dimensional case yields:

\[
\Delta x^0 = -J^{-1}(x^0) \cdot f(x^0);
\] (16.5)

thus, by defining \( x^1 := x^0 + \Delta x^0 \), we have

\[
x^1 = x^0 - J^{-1}(x^0) \cdot f(x^0).
\] (16.6)

- The iteration scheme is then clear:

\[
x^{\nu+1} = x^{\nu} - J^{-1}(x^{\nu}) \cdot f(x^{\nu})
\] (16.7)

for some \( x^0 \in \mathbb{R}^n \).

- The algorithm terminates when \( \|x^{\nu+1} - x^{\nu}\|_2 < \varepsilon \), for some \( \varepsilon > 0 \) small.

When executing the iteration in (16.7), one does not compute the inverse of the Jacobian, \( J^{-1}(x^{\nu}) \) because it is computationally expensive to do so. Instead, at each step \( \nu \), given \( x^{\nu} \), one solves for \( \Delta x^{\nu} \) in the following linear system of equations:

\[
J(x^{\nu}) \Delta x^{\nu} = -f(x^{\nu}),
\] (16.8)

and updates the current estimate of the solution to \( f(x) = 0 \), \( x^{\nu} \), as follows:

\[
x^{\nu+1} = x^{\nu} + \Delta x^{\nu}.
\]

Next, we discuss how to use the N-R algorithm to solve the power flow problem for the cases discussed earlier:

**C1.** System with 1 generator and \( n - 1 \) loads.

**C2.** System with \( 1 < m < n \) generators and \( n - m \) loads.
Case I: System with 1 generator and \( n - 1 \) loads

We first discuss the particular case of a two-bus system with one generator and one load and then extend the ideas to the case of a system with \( n \) buses \( (n > 2) \) with 1 generator and \( n - 1 \) loads.

![Two-Bus Case](image)

Two-Bus Case

Recall two-bus example discussed earlier (see Fig. 16.2 for the one-line diagram):

- Given: \( V_1 = E_1, \theta_1 = 0, P_2 = -P_2^D, Q_2 = -Q_2^D \).
- To compute: \( P_1, Q_1, \theta_2, V_2 \).

Bus 1 active and reactive power balance equations (slightly rewritten):

\[
\frac{E_1V_2}{X_{12}} \sin(-\theta_2) - P_1 = 0, \quad \text{=: } p_1(\theta_2, V_2)
\]

\[
\frac{E_1^2}{X_{12}} - \frac{E_1V_2}{X_{12}} \cos(-\theta_2) - Q_1 = 0. \quad \text{=: } q_1(\theta_2, V_2)
\]
Bus 2 active and reactive power balance equations (slightly rewritten):

\[
\begin{align*}
\frac{V_1 V_2}{X_{12}} \sin(\theta_2) + P_2^D &= 0, \\
\frac{V_2^2}{X_{12}} - \frac{E_1 V_2}{X_{12}} \cos(\theta_2) + Q_2^D &= 0.
\end{align*}
\]

As discussed earlier, the active and reactive power balance equations for bus 2 form a closed set of equations with 2 equations and 2 unknowns, \(\theta_2, V_2\); thus, we can solve for these first, and then used the computed values in the active and reactive power balance equations for bus 1 to compute the value of the two other unknowns, \(P_1, Q_1\). This implies that when setting up the N-R iteration, we only need to use the active and reactive power balance equation for bus 2; thus, we have

\[
\begin{bmatrix}
\theta_2^{n+1} \\
V_2^{n+1}
\end{bmatrix} =
\begin{bmatrix}
\theta_2^n \\
V_2^n
\end{bmatrix} - \begin{bmatrix}
\frac{\partial p_2}{\partial \theta_2} |_{\theta_2^n, V_2^n} & \frac{\partial q_2}{\partial \theta_2} |_{\theta_2^n, V_2^n} \\
\frac{\partial p_2}{\partial V_2} |_{\theta_2^n, V_2^n} & \frac{\partial q_2}{\partial V_2} |_{\theta_2^n, V_2^n}
\end{bmatrix}^{-1} \begin{bmatrix}
p_2(\theta_2^n, V_2^n) + P_2^D \\
q_2(\theta_2^n, V_2^n) + Q_2^D
\end{bmatrix}.
\]

General Case: System With \(n\) Buses \((n > 2)\)

The setting in this case is as follows:

- Given: \(\theta_1 = 0, V_1 = E_1, P_i = -P_i^D, i = 2, \ldots, n,\) and \(Q_i = -Q_i^D, i = 2, \ldots, n.\)
- To compute: \(P_1, Q_1, \theta_i, i = 2, \ldots, n,\) and \(V_i, i = 2, \ldots, n.\)

As discussed earlier, if one considers the active and reactive power balance equations for buses \(i = 2, 3, \ldots, n,\) one obtains \(2n - 2\) equations with each involving the following unknowns: \(\theta_2, V_2, \theta_3, V_3, \ldots, \theta_n, V_n\) \((2n - 2\) unknowns); thus these equations form a closed system. Then, it is possible to solve this “reduced” system of equations for the remaining unknowns, and utilize the results obtained to compute the value of \(P_1\) and \(Q_1\) after substitution in the active and reactive power balance equations for bus 1. The procedure is summarized as follows:

\[\text{S1} \quad \text{Consider the active and reactive power balance equations (slightly rewritten) for load buses, i.e.,}
\]

\[
\begin{align*}
\sum_{k=1}^{n} V_i V_k \left( G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k) \right) + P_i^D &= 0, \quad i = 2, \ldots, n, \quad (16.9) \\
\sum_{k=1}^{n} V_i V_k \left( G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k) \right) + Q_i^D &= 0, \quad i = 2, \ldots, n, \quad (16.10)
\end{align*}
\]

with \(\theta_1 = 0\) and \(V_1 = E_1,\) and solve for \(\theta_2, \ldots, \theta_n\) and \(V_2, \ldots V_n.\)
S2 Use the values computed in Step S1, which we denote by \( \theta_2^*, \ldots, \theta_n^* \) and \( V_2^*, \ldots, V_n^* \), and substitute them into the active and reactive power balance equation for bus 1 to compute the values that \( P_1 \) and \( Q_1 \) take, i.e.

\[
P_1 = \sum_{k=1}^{n} V_1 V_k \left( G_{1k} \cos(\theta_i - \theta_k) + B_{1k} \sin(\theta_i - \theta_k) \right), \tag{16.11}
\]

\[
Q_1 = \sum_{k=1}^{n} V_1 V_k \left( G_{1k} \sin(\theta_i - \theta_k) - B_{1k} \cos(\theta_i - \theta_k) \right), \tag{16.12}
\]

with \( \theta_1 = 0 \) and \( V_1 = E_1 \), and \( \theta_2 = \theta_2^*, \ldots, \theta_n^* \) and \( V_2 = V_2^*, \ldots, V_n = V_n^* \).

Thus, when setting up the Newton-Raphson iteration, we only need to consider (16.9) – (16.10).

Define \( x := [\theta_2, \theta_3, \ldots, \theta_n, V_2, V_3, \ldots, V_n]^\top \) and

\[
p_i(x) = \sum_{k=1}^{n} V_i V_k \left( G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k) \right), \tag{16.13}
\]

and

\[
q_i(x) = \sum_{k=1}^{n} V_i V_k \left( G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k) \right). \tag{16.14}
\]

Then, the N-R iteration for (16.9) – (16.10) is:

\[
x^{\nu+1} = x^{\nu} - (J(x^{\nu}))^{-1} f(x^{\nu}), \tag{16.15}
\]
where

\[
J(x^\nu) = \begin{bmatrix}
\frac{\partial p_2}{\partial \theta_2} |_{x^\nu} & \cdots & \frac{\partial p_2}{\partial \theta_n} |_{x^\nu} & \frac{\partial p_2}{\partial V_2} |_{x^\nu} & \cdots & \frac{\partial p_2}{\partial V_n} |_{x^\nu} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial q_2}{\partial \theta_2} |_{x^\nu} & \cdots & \frac{\partial q_2}{\partial \theta_n} |_{x^\nu} & \frac{\partial q_2}{\partial V_2} |_{x^\nu} & \cdots & \frac{\partial q_2}{\partial V_n} |_{x^\nu} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial q_n}{\partial \theta_2} |_{x^\nu} & \cdots & \frac{\partial q_n}{\partial \theta_n} |_{x^\nu} & \frac{\partial q_n}{\partial V_2} |_{x^\nu} & \cdots & \frac{\partial q_n}{\partial V_n} |_{x^\nu}
\end{bmatrix},
\]

\[
f(x^\nu) = \begin{bmatrix}
p_2(x^\nu) + P_2^D \\
\vdots \\
p_n(x^\nu) + P_n^D \\
\vdots \\
q_2(x^\nu) + Q_2^D \\
\vdots \\
q_n(x^\nu) + Q_n^D
\end{bmatrix},
\]

for some given \( x^0 = [\theta_2^0, \theta_3^0, \ldots, \theta_n^0, V_2^0, V_3^0, \ldots, V_n^0]^\top \).

**Initialization.** The solution to the N-R iteration will depend on the choice of \( x^0 = [\theta_2^0, \theta_3^0, \ldots, \theta_n^0, V_2^0, V_3^0, \ldots, V_n^0]^\top \).

In normal conditions, a power system is expected to operate with a voltage profile that is close to the so-called flat voltage profile, i.e., \( \theta_i = 0 \) and \( V_i = 1 \) p.u., for all \( i = 1, 2, \ldots, n \); thus a natural choice for \( x^0 \) is such flat voltage profile, i.e.,

\[
x^0 = [\theta_2^0, \theta_3^0, \ldots, \theta_n^0, V_2^0, V_3^0, \ldots, V_n^0]^\top,
\]

where \( \theta_i^0 = 0 \) and \( V_i = 0 \) for all \( i = 2, \ldots, n \).

**Executing the N-R iteration in practice.** As discussed earlier, when executing (16.15), we do not compute the inverse of the Jacobian because for a large system this would be computationally expensive; instead we would solve for \( \Delta x^\nu \) in

\[
J(x^\nu)\Delta x^\nu = -f(x^\nu),
\]

and update the value of \( x^\nu \) to \( x^{\nu+1} \) as follows:

\[
x^{\nu+1} = x^\nu + \Delta x^\nu.
\]