POWER FLOW ANALYSIS

- Power flow analysis assumption
  - steady-state
  - balanced single-phase network
  - network may contain hundreds of nodes and branches with impedance X specified in per unit on MVA base
- Power flow equations
  - bus admittance matrix of node-voltage equation is formulated
  - currents can be expressed in terms of voltages
  - resulting equation can be in terms of power in MW

BUS ADMITTANCE MATRIX

- Nodal solution
  - nodal solution is based on the Kirchhoff's current law
  - impedance is converted to admittance

$$y_{ij} = \frac{1}{Z_{ij}} = \frac{1}{r_{ij} + jx_{ij}}$$

- Bus admittance equations
  - the impedance diagram: *see Fig.6.1*

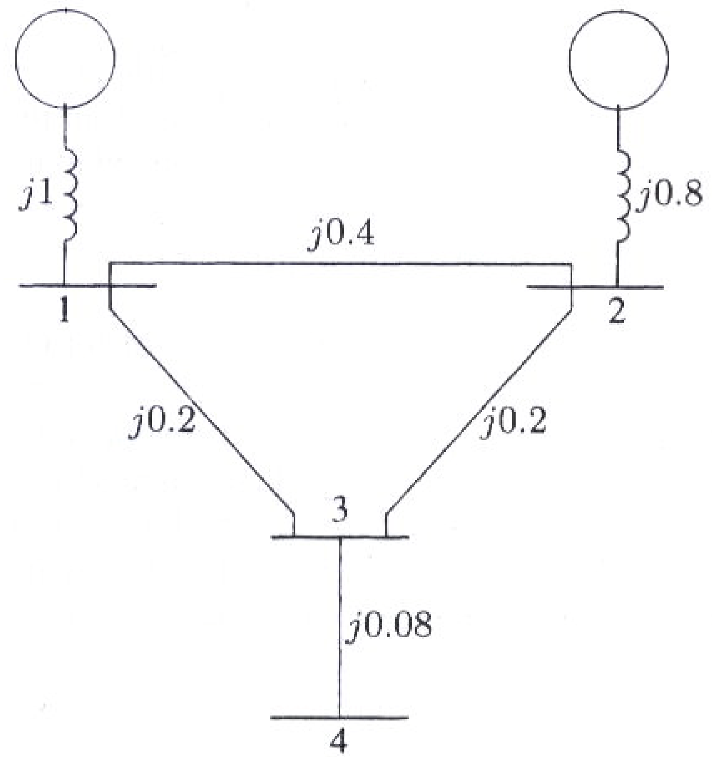


FIGURE 6.1
The impedance diagram of a simple system.

BUS ADMITTANCE MATRIX

- Bus admittance equations
  - the admittance is based on bus-to-bus: *see Fig.6.2*
  - if no connection between bus-to-bus, leave as zero
  - node voltage equation is in the form

$$I_{bus} = Y_{bus} V_{bus}$$

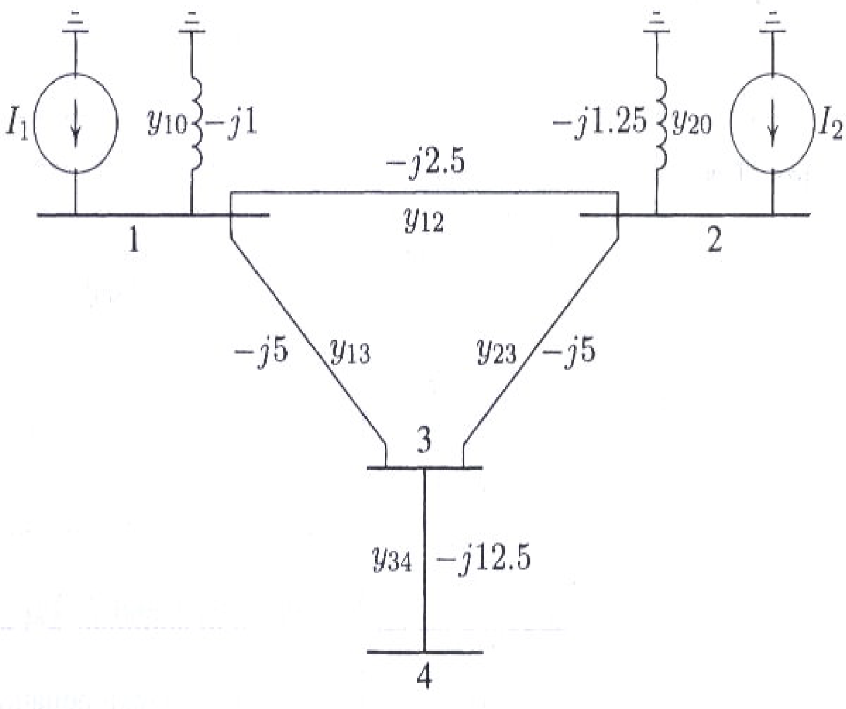


FIGURE 6.2
The admittance diagram for system of Figure 6.1.

BUS ADMITTANCE MATRIX

- Node-voltage matrix
  - $I_{bus} = Y_{bus} V_{bus}$

$$
\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_i \\ I_n \end{bmatrix} =
\begin{bmatrix}
\mathbf{Y_{11}} & \mathbf{Y_{12}} & \cdots & \mathbf{Y_{1i}} & \cdots & \mathbf{Y_{1n}} \\
\mathbf{Y_{21}} & \mathbf{Y_{22}} & \cdots & \mathbf{Y_{2i}} & \cdots & \mathbf{Y_{2n}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{Y_{i1}} & \mathbf{Y_{i2}} & \cdots & \mathbf{Y_{ii}} & \cdots & \mathbf{Y_{in}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{Y_{n1}} & \mathbf{Y_{n2}} & \cdots & \mathbf{Y_{ni}} & \cdots & \mathbf{Y_{nn}}
\end{bmatrix}
\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_i \\ V_n \end{bmatrix}
$$

  - I_{bus} is the vector of injected currents
  - V_{bus} is the vector of the bus voltage from reference node
  - Y_{bus} is the bus admittance matrix

BUS ADMITTANCE MATRIX

- Node-voltage matrix
  - diagonal element Y_{ii}: sum of admittance connected to bus i

$$Y_{ii} = \sum_{j=0}^{n} y_{ij} \qquad j \neq i$$

  - off-diagonal matrix Y_{ij}: negative of admittance between nodes I and j

$$Y_{ij} = Y_{ji} = -y_{ij}$$

  - when the bus currents are known, bus voltages are unknown, bus voltage can be solved as

$$V_{bus} = Y_{bus}^{-1} I_{bus}$$

  - inverse of bus admittance matrix is known as impedance matrix Z_{bus}

$$Z_{bus} = Y_{bus}^{-1}$$

  - if matrix of Y_{bus} is invertible, Y_{bus} should be non-singular

BUS ADMITTANCE MATRIX

- Node-voltage matrix
  - admittance matrix is symmetric along the leading diagonal, which result in an upper diagonal nodal admittance matrix
  - a typical power system network, each bus is connected by a few nearby bus, which cause many off-diagonal elements are zero
  - many zero off-diagonal matrix is called sparse matrix
  - the bus admittance matrix in Fig.(6.2) by inspection is

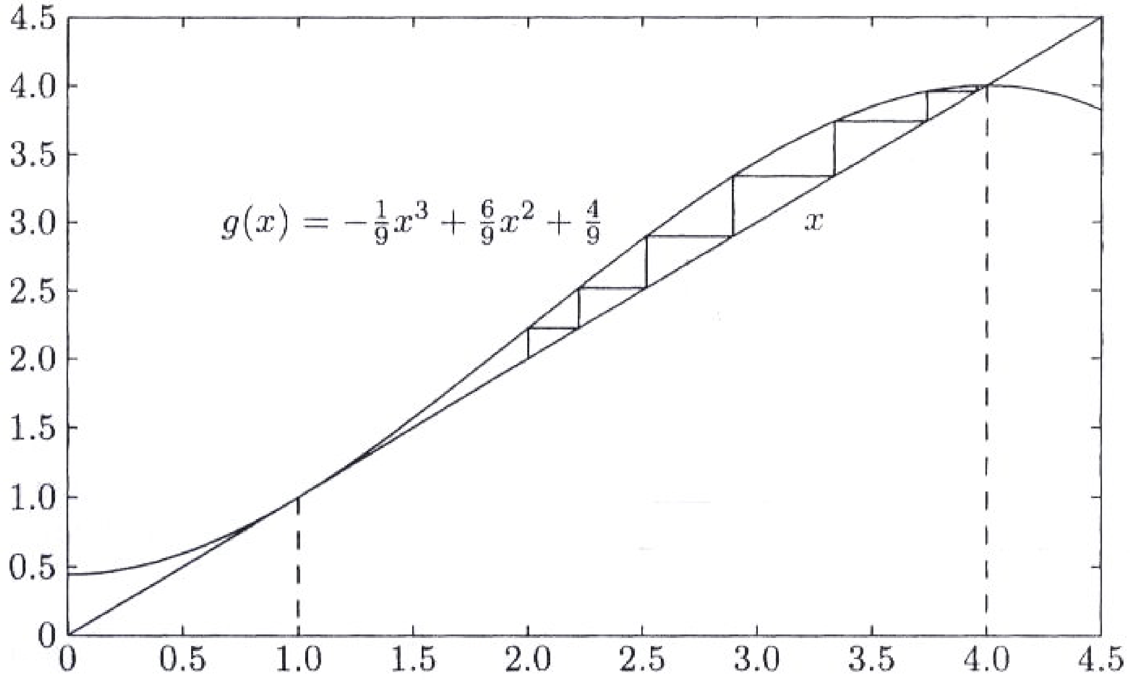
$$Y_{bus} = \begin{bmatrix} -j8.5 & j2.5 & j5.0 & 0 \\ j2.5 & -j8.75 & j5.0 & 0 \\ j5.0 & j5.0 & -j22.5 & j12.5 \\ 0 & 0 & j12.5 & -j12.5 \end{bmatrix}$$

SOLUTION OF NONLINEAR ALGEBRA EQUATIONS

- Techniques for iterative solution of non-linear equations
  - Gauss-Seidal
  - Newton-Raphson
  - Quasi-Newton
- Gause-Seidal method
  - consider a nonlinear equation $f(x)=0$
  - rearrange $f(x)$ so that $x=g(x)$, $f(x)=x-g(x)$ or $f(x)=g(x)-x$
  - guess an initial estimate of $x = x^{(k)}$
  - use iteration, obtain next x value as $x^{(k+1)} = g(x^{(k)})$
  - criteria for stop iteration: $|x^{(k+1)}-x^{(k)}| \leq \varepsilon$
  - ε is the desired accuracy

GAUSE-SEIDAL METHOD

- Nature of Gause-Seidal method
  - *see Ex.(6.2) and Fig.(6.3)*
  - Gause-Seidal method needs many iterations to achieve desired accuracy
  - no guarantee for the convergence, depend on the location of initial x estimate



GAUSE-SEIDAL METHOD

- Nature of Gause-Seidal method
  - solution: if initial estimate x is within convergent region, solution will converge in zigzag fashion to one of the roots
  - no solution: if initial estimate x is outside convergent region, process will diverge, no solution found
  - in some case, an acceleration factor α is added to improve the rate of convergence:
  - $x^{(k+1)} = x^{(k)} + \alpha[g(x^{(k)}) - x^{(k)}]$, where $\alpha > 1$
  - acceleration factor should not too large to produce overshoot
  - *see Ex.(6.3)* for the acceleration factor used

GAUSE-SEIDAL METHOD

- Extend one variable to n variable equations using Gause-Seidal method
  - consider the system of n equations in n variables and solving for one variable from each equation in one time of iteration

$$
\begin{aligned}
f_1(x_1, x_2, \cdots, x_n) &= c_1 & \qquad x_1 &= c_1 + g_1(x_1, x_2, \cdots, x_n) \\
f_2(x_1, x_2, \cdots, x_n) &= c_2 & \qquad x_2 &= c_2 + g_2(x_1, x_2, \cdots, x_n) \\
\cdots\cdots\cdots\cdots & & \qquad \cdots\cdots\cdots\cdots & \\
f_n(x_1, x_2, \cdots, x_n) &= c_n & \qquad x_n &= c_n + g_n(x_1, x_2, \cdots, x_n)
\end{aligned}
$$

  - the updated variable $x_1^{(k+1)}$ calculated in first equation in Eq.(6.12) is used in the calculation of $x_2^{(k+1)}$ in the second equation
  - Ex: in the 2nd iteration $x_2^{(k+1)} = c_2+g_2(x_1^{(k+1)}+x_2^{(k)}+x_3^{(k)}+\ldots+x_n^{(k)})$
  - at n iteration to complete n variables, the $x_1^{(k+1)},\ldots,x_n^{(k+1)}$ is tested against $x_1^{(k)},\ldots,x_n^{(k)}$ for accuracy criterion

POWER FLOW SOLUTION

- Power Flow (Load Flow)
  - operating condition: balanced, single phase model
  - quantities used in power flow equation are: voltage magnitude $|V|$, phase angle δ, real power P, and reactive power Q
  - system bus classification:
    - slack bus (swing bus): taken as reference where $|V|$ and $\angle V$ are specified. It makes up the loss between generated power and scheduled loads
    - load bus (PQ bus): P and Q are specified, $|V|$ and $\angle V$ are unknown
    - regulated bus (PV bus): P and $|V|$ are specified, $\angle V$ and Q are unknown

POWER FLOW EQUATION

- Power flow formulation
  - consider bus case in Fig.(6.7)
  - current flow into bus i:

$$I_i = V_i \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij} V_j \qquad j \neq i$$

  - express I_i in terms of P,Q:

$$I_i = \frac{P_i - jQ_i}{V_i^*}$$

  - the power flow equation becomes

$$\frac{P_i - jQ_i}{V_i^*} = V_i \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij} V_j \qquad j \neq i$$

  - the power flow problem results in algebraic nonlinear equations which must be solved by iteration methods

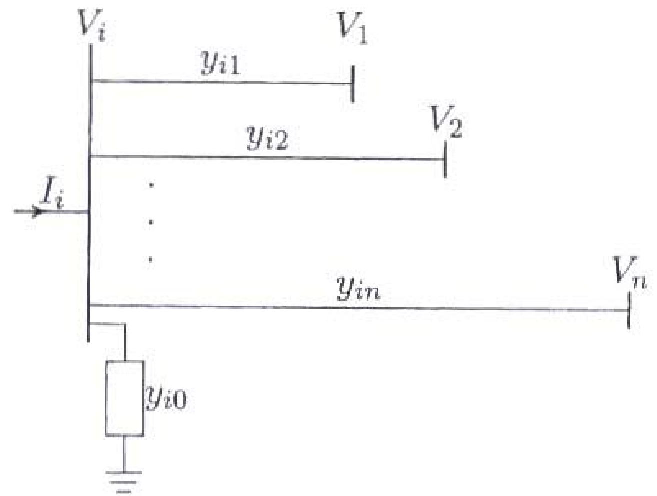


FIGURE 6.7
A typical bus of the power system.

GAUSS-SEIDEL POWER FLOW EQUATION

- Gauss-Seidel power flow solution
  - solving V_i: for PQ bus, assume P,Q are known

$$V_i^{(k+1)} = \frac{\dfrac{P_i^{sch} - jQ_i^{sch}}{V_i^{*(k)}} + \sum y_{ij} V_k^{(k)}}{\sum y_{ij}} \qquad \mathrm{j \neq i}$$

  - solving P_i: for slack bus, assume V is known

$$P_i^{(k+1)} = \mathrm{Re}\left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} V_j^{(k)} \right] \right\} \qquad \mathrm{j \neq i}$$

  - solving Q_i: for PV bus, assume |V| is known

$$Q_i^{(k+1)} = -\mathrm{Im}\left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} V_j^{(k)} \right] \right\} \qquad \mathrm{j \neq i}$$

GAUSS-SEIDEL POWER FLOW EQUATION

- Instructions for Gauss-Seidel solution
  - there are 2(n-1) equations to be solved for n bus
  - voltage magnitude of the buses are close to 1pu or close to the magnitude of the slack bus
  - voltage magnitude at load buses is lower than the slack bus value
  - voltage magnitude at generator buses is higher than the slack bus value
  - phase angle of load buses are below the reference angle
  - phase angle of generator buses are above the reference angle

INSTRUCTIONS FOR G-S SOLUTION

- Instructions for PQ bus solution
  - real and reactive power P_i^{sch}, Q_i^{sch} are known
  - starting with an initial estimate of voltage using V_i equation

$$V_i^{(k+1)} = \frac{\dfrac{P_i^{sch} - jQ_i^{sch}}{V_i^{*(k)}} + \sum y_{ij} V_j^{(k)}}{\sum y_{ij}} \qquad j \neq i$$

- Instructions for PV bus solution
  - P_i^{sch}, $|V_i|$ are specified
  - assume $V_i = |V_i|\angle 0^o$, solve the Q_i equation as below

$$Q_i^{(k+1)} = -\operatorname{Im}\left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} V_j^{(k)} \right] \right\} \qquad j \neq i$$

INSTRUCTIONS FOR G-S SOLUTION

- Instructions for PV bus solution
  - when $Q_i^{(k+1)}$ is available, solve V_i using equation below

$$V_i^{(k+1)} = \frac{\dfrac{P_i^{sch} - jQ_i^{(k)}}{V_i^{*(k)}} + \sum y_{ij} V_k^{(k)}}{\sum y_{ij}} \qquad j \neq i$$

  - since $|V_i|$ is specified, keep imaginary part of V_i, calculate real part of V_i

$$\mathrm{Re}\{V_i^{(k+1)}\} = \sqrt{|V_i|^2 - \left(imag\{V_i^{(k+1)}\}\right)^2}$$

  - solve V_i

$$V_i^{(k+1)} = \mathrm{Re}\{V_i^{(k+1)}\} + j\,\mathrm{Im}\{V_i^{(k+1)}\}$$

  - stopping criteria

$$\left|\mathrm{Re}\{V_i^{(k+1)}\} - \mathrm{Re}\{V_i^{(k)}\}\right| \le \varepsilon, \quad \left|\mathrm{Im}\{V_i^{(k+1)}\} - \mathrm{Im}\{V_i^{(k)}\}\right| \le \varepsilon$$

INSTRUCTIONS FOR G-S SOLUTION

- Instructions for PV bus solution
  - to accelerate the convergence, using the following approximation after new V_i is obtained

$$V_i^{(k+1)} = V_i^{(k)} + \alpha\left(V_{i\,cal}^{(k)} - V_i^{(k)}\right)$$

  - α is in the range between 1.3 to 1.7
  - voltage accuracy in $|V_i|$ and $\angle\delta$ is in the range between 0.00001 to 0.00005

INSTRUCTIONS FOR G-S SOLUTION

- Instructions for V, $\angle\delta$ slack bus solution
  - solve P_i

$$P_i^{(k+1)} = \mathrm{Re}\left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} V_j^{(k)} \right] \right\} \qquad j \neq i$$

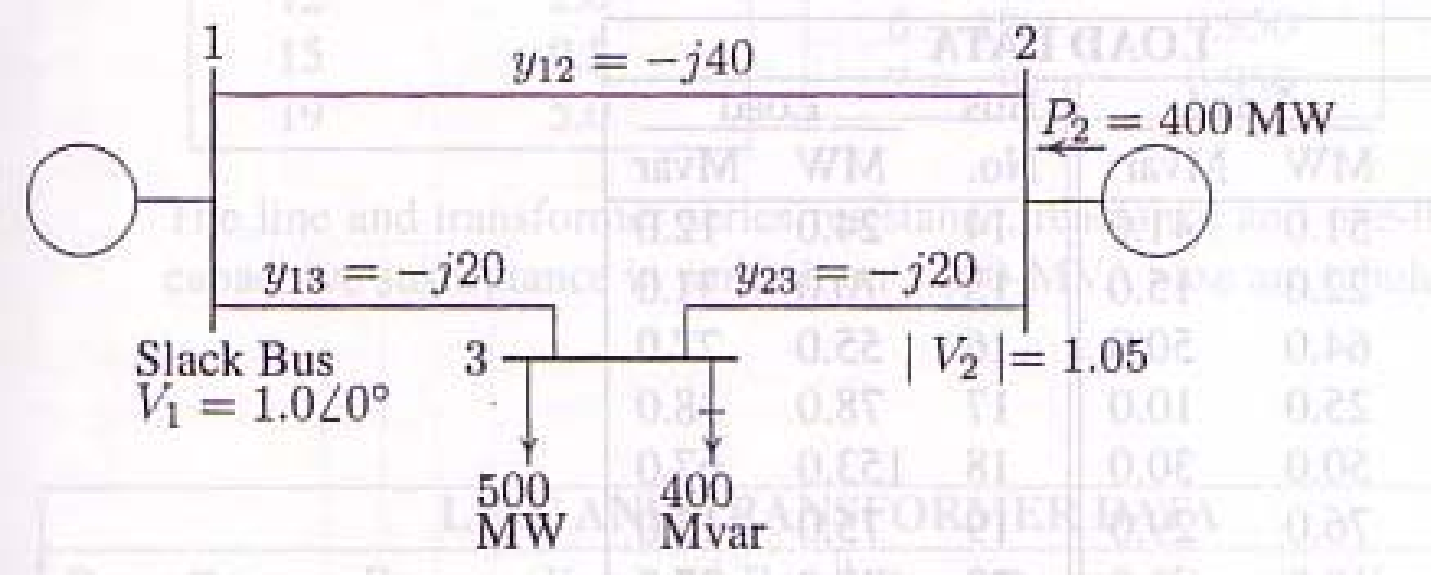
  - solve Q_i

$$Q_i^{(k+1)} = -\mathrm{Im}\left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} V_j^{(k)} \right] \right\} \qquad j \neq i$$

  - accuracy: the largest ΔPΔQ is less than the specified value, typically is about 0.001pu

G-S Power flow Homework

For the one-line diagram shown below, using the G-S method to determine all bus voltages (magnitude and phase) and show the power flow solution between the buses assume the regulated bus (#2) reactive power limits are between 0 and 600Mvar.



NEWTON RAPHSON METHOD

- Newton Raphson method for solving one variable
  - consider the solution of one-dimensional equation $f(x)=c$
  - assume $x = x^{(0)} + \Delta x^{(0)}$
  - $f(x)=f(x^{(0)}+\Delta x^{(0)})=c$
  - use Taylor's series expansion

$$f(x^{(0)}+\Delta x^{(0)}) = f(x^{(0)}) + \left(\frac{df}{dx}\right)^{(0)} \Delta x^{(0)} + \frac{1}{2!}\left(\frac{d^2 f}{dx^2}\right)^{(0)} \left(\Delta x^{(0)}\right)^2 + \cdots = c$$

  - assume $\Delta x^{(0)}$ is very small, higher order terms of expansion can be neglected, Taylor series becomes

$$f(x^{(0)}+\Delta x^{(0)}) = f(x^{(0)}) + \left(\frac{df}{dx}\right)^{(0)} \Delta x^{(0)} = c$$

  - assume $f(x^{(0)})=c-\Delta c^{(0)}$, the equation becomes $\Delta c^{(0)} \cong (df/dx)^{(0)} \Delta x^{(0)}$
  - the new approximation of x $\quad x^{(1)} = x^{(0)} + \dfrac{\Delta c^{(0)}}{\left(\dfrac{df}{dx}\right)^{(0)}}$

NEWTON RAPHSON METHOD

- Newton Raphson method for solving one variable
  - the new approximation of x

$$x^{(1)} = x^{(0)} + \frac{\Delta c^{(0)}}{\left(\frac{df}{dx}\right)^{(0)}}$$

- Newton Raphson algorithm
  - $$\Delta c^{(k)} = c - f(x^{(k)})$$

$$\Delta x^{(k)} = \frac{\Delta c^{(k)}}{\left(\frac{df}{dx}\right)^{(k)}}$$

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

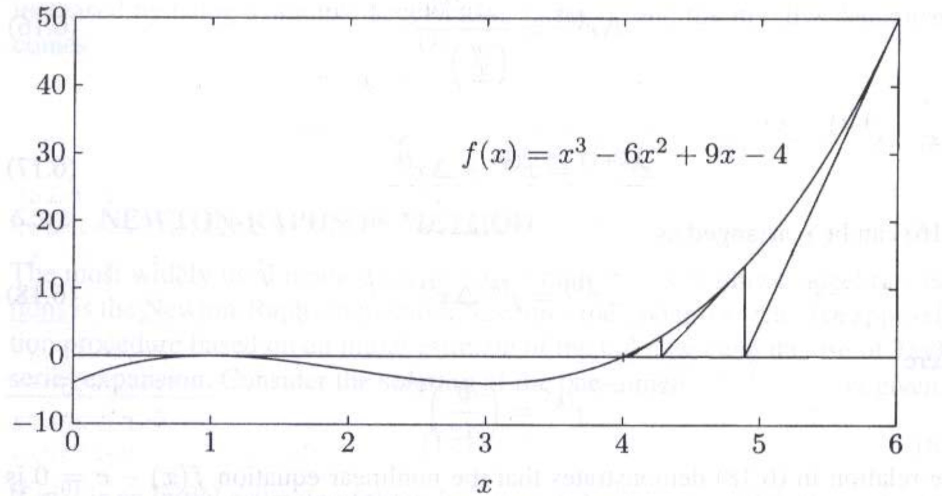


FIGURE 6.5
Graphical illustration of the Newton-Raphson algorithm.

  - for more information, *see Ex.(6.4)*
  - Newton's method converges faster than Gauss-Seidal, the root may converge to a root different from the expected one or diverge if the starting value is not close enough to the root

NEWTON RAPHSON METHOD FOR n VARIABLES

- Newton Raphson method for solving n variables

$$f_1(x^{(0)} + \Delta x^{(0)}) = f_1(x^{(0)}) + \left(\frac{\partial f_1}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial f_1}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial f_1}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_1$$

$$f_2(x^{(0)} + \Delta x^{(0)}) = f_2(x^{(0)}) + \left(\frac{\partial f_2}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial f_2}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial f_2}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_2$$

$$f_n(x^{(0)} + \Delta x^{(0)}) = f_n(x^{(0)}) + \left(\frac{\partial f_n}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial f_n}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial f_n}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_n$$

NEWTON RAPHSON METHOD FOR n VARIABLES

- Rearrange in matrix form

$$\begin{bmatrix} c_1 - f_1^{(0)} \\ c_2 - f_2^{(0)} \\ \vdots \\ c_n - f_n^{(0)} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)^{(0)} & \left(\frac{\partial f_1}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial f_1}{\partial x_n}\right)^{(0)} \\ \left(\frac{\partial f_2}{\partial x_1}\right)^{(0)} & \left(\frac{\partial f_2}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial f_2}{\partial x_n}\right)^{(0)} \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)^{(0)} & \left(\frac{\partial f_n}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial f_n}{\partial x_n}\right)^{(0)} \end{bmatrix} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \vdots \\ \Delta x_n^{(0)} \end{bmatrix}$$

- The matrix can be written as
  - $\Delta C^{(k)} = J^{(k)} \Delta X^{(k)}$

NEWTON RAPHSON METHOD FOR n VARIABLES

- The Newton-Raphson algorithm for n-dimensional case is
  - $X^{(k+1)} = X^{(k)} + \Delta X^{(k)} = X^{(k)} + [J^{(k)}]^{-1}\Delta C^{(k)}$
  - where

$$\Delta X^{(k)} = \begin{bmatrix} \Delta x_1^{(k)} \\ \Delta x_2^{(k)} \\ \vdots \\ \Delta x_n^{(k)} \end{bmatrix}$$

$$\Delta C^{(k)} = \begin{bmatrix} c_1 - f_1^{(k)} \\ c_2 - f_2^{(k)} \\ \vdots \\ c_n - f_n^{(k)} \end{bmatrix}$$

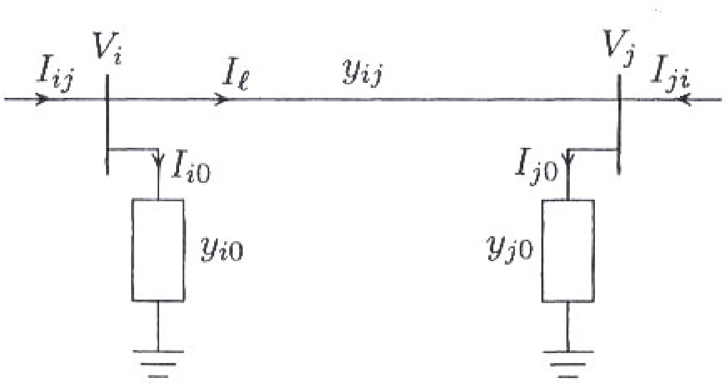
$$J^{(k)} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)^{(k)} & \left(\frac{\partial f_1}{\partial x_2}\right)^{(k)} & \cdots & \left(\frac{\partial f_1}{\partial x_n}\right)^{(k)} \\ \left(\frac{\partial f_2}{\partial x_1}\right)^{(k)} & \left(\frac{\partial f_2}{\partial x_2}\right)^{(k)} & \cdots & \left(\frac{\partial f_2}{\partial x_n}\right)^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)^{(k)} & \left(\frac{\partial f_n}{\partial x_2}\right)^{(k)} & \cdots & \left(\frac{\partial f_n}{\partial x_n}\right)^{(k)} \end{bmatrix}$$

NEWTON RAPHSON METHOD FOR n VARIABLES

- The Newton-Raphson algorithm
  - $J^{(k)}$ is called the Jacobian matrix
  - solution to $X^{(k+1)}$ is inefficient because it involves inverse of $J^{(k)}$, a triangular factorization is used to facilitate the computation
  - in MATLAB, the operator “\” (i.e., $\Delta X = J \backslash \Delta C$) is used to apply the triangular factorization
  - Newton-Raphson method converge to solution quadratically when near a root
  - The limitation is that it does not generally converge to a solution from an arbitrary starting point

LINE FLOWS AND LOSSES

- Complex power flow between bus i,j
  - for line model, *see Fig. 6.8*
  - current flow from bus i to bus j



$$I_{ij} = I_l + I_{i0} = y_{ij}(V_i - V_j) + y_{i0}V_i$$

  - current flow from bus j to bus i

$$I_{ji} = -I_l + I_{j0} = y_{ij}(V_j - V_i) + y_{j0}V_j$$

  - complex power S_{ij} from bus i to j and S_{ji} from j to i

$$S_{ij} = V_i I_{ij}^* \qquad S_{ji} = V_j I_{ji}^*$$

  - power loss in the line i-j

$$S_{L(i-j)} = S_{ij} + S_{ji}$$

  - for more Gauss-Seidel method examples, see Ex. (6.7) and Ex. (6.8)

NEWTON-RAPHSON POWER FLOW

- Real power flow in terms of V_i , $\angle\delta$, and Y_{ij}

$$P_i = \sum_{j=1}^{n} |V_i||V_j||Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j)$$

- Reactive power flow

$$Q_i = -\sum_{j=1}^{n} |V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j)$$

- Newton-Raphson matrix form: $\Delta C^{(k)} = J^{(k)} \Delta X^{(k)}$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta|V| \end{bmatrix}$$

  - diagonal and off-diagonal elements of J_1

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{j \neq i} |V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j)$$

$$\frac{\partial P_i}{\partial \delta_j} = -|V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) \qquad j \neq i$$

NEWTON-RAPHSON POWER FLOW

- Newton-Raphson matrix form: $\Delta C^{(k)} = J^{(k)} \Delta X^{(k)}$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

  - diagonal and off-diagonal elements of J_2

$$\frac{\partial P_i}{\partial |V_i|} = 2|V_i||Y_{ii}|\cos\theta_{ii} + \sum_{j \neq i} |V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j)$$

$$\frac{\partial P_i}{\partial |V_j|} = |V_i||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j) \quad \mathrm{j} \neq \mathrm{i}$$

  - diagonal and off-diagonal elements of J_3

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{j \neq i} |V_i||V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j)$$

$$\frac{\partial Q_i}{\partial \delta_j} = -|V_i||V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j) \quad \mathrm{j} \neq \mathrm{i}$$

NEWTON-RAPHSON POWER FLOW

- Newton-Raphson matrix form: $\Delta C^{(k)} = J^{(k)} \Delta X^{(k)}$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

  - diagonal and off-diagonal elements of J_4

$$\frac{\partial Q_i}{\partial |V_i|} = -2|V_i||Y_{ii}|\sin\theta_{ii} - \sum_{j \neq i} |V_j||Y_{ij}|\sin(\theta_{ij} - \delta_i + \delta_j)$$

$$\frac{\partial Q_i}{\partial |V_j|} = -|V_i||Y_{ij}|\sin(\theta_{ij} - \delta_i + \delta_j) \qquad j \neq i$$

  - power residuals $\Delta P_i^{(k)}$ $\Delta Q_i^{(k)}$

$$\Delta P_i^{(k)} = P_i^{sch} - P_i^{(k)}, \qquad \Delta Q_i^{(k)} = Q_i^{sch} - Q_i^{(k)}$$

  - new estimates for bus voltages

$$\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta\delta_i^{(k)}, \qquad |V_i^{(k+1)}| = |V_i^{(k)}| + |\Delta V_i^{(k)}|$$

NEWTON-RAPHSON POWER FLOW

- Procedure for Newton-Raphson method:
  - PQ bus: set $|V_i^{(0)}|=1.0$, $\delta_i^{(0)}=0.0$
  - PV bus: set $\delta_i^{(0)}=0.0$
  - set PQ bus equation for J matrix elements:

$$P_i = \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j)$$

$$Q_i = -\sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\sin(\theta_{ij} - \delta_i + \delta_j)$$

$$\Delta P_i^{(k)} = P_i^{sch} - P_i^{(k)}, \qquad \Delta Q_i^{(k)} = Q_i^{sch} - Q_i^{(k)}$$

  - set PV bus equation for J matrix elements:

$$P_i = \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\cos(\theta_{ij} - \delta_i + \delta_j)$$

$$\Delta P_i^{(k)} = P_i^{sch} - P_i^{(k)}$$

NEWTON-RAPHSON POWER FLOW

- Procedure for Newton-Raphson method:
  - use above equation to calculate Jacobian matrix (J_1, J_2, J_3, J_4)
  - solve $\Delta|V|$ and $\Delta\delta$ using Newton-Raphson matrix

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta|V| \end{bmatrix}$$

  - update $\Delta|V|$ and $\Delta\delta$ by

$$\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta\delta_i^{(k)}, \qquad \left|V_i^{(k+1)}\right| = \left|V_i^{(k)}\right| + \left|\Delta V_i^{(k)}\right|$$

  - repeat the calculation until

$$\left|\Delta P_i^{(k)}\right| \le \varepsilon, \quad \left|\Delta Q_i^{(k)}\right| \le \varepsilon$$

  - for example: *see Ex.(6.10)*

FAST DECOUPLED POWER FLOW

- Fast decoupled power flow solution:
  - the algorithm is based on Newton-Raphson method
  - when transmission lines has a high X/R ratio, the Newton-Raphson method could be further simplified
- Consider the Newton-Raphson power flow equation
  - $$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$
  - ΔP are less sensitive to $|V|$ and most sensitive to $\Delta\delta$
  - ΔQ is less sensitive to $\Delta\delta$ and most sensitive to $|V|$
  - we can reasonably eliminate J_2 and J_3 elements in Jacobian matrix

FAST DECOUPLED POWER FLOW

- Consider the Newton-Raphson power flow equation
  - the power flow equation reduces to

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

  - $\Delta P = J_1 \Delta\delta = [\partial P/\partial\delta]\Delta\delta$, $\Delta Q = J_4 \Delta|V| = [\partial Q/\partial|V|]\Delta|V|$
  - $\partial P_i/\partial\delta_i = -Q_i - |V_i|^2 B_{ii}$, $B_{ii} = |Y_{ii}|\sin\theta_{ii}$ is the imaginary part of the diagonal elements
  - since $B_{ii} >> Q_i$, $\partial P_i/\partial\delta_i$ (diagonal elements of J_1) can be further reduced to $\partial P_i/\partial\delta_i = -|V_i|B_{ii}$ ($|V_i|^2 \approx |V_i|$)
  - off diagonal element of J_1: $\partial P_i/\partial\delta_j = -|V_i||V_j|Y_{ij}\sin(\theta_{ij}-\delta_i+\delta_j)$, since $\delta_j-\delta_i$ is quite small, $\theta_{ij}-\delta_i+\delta_j = \theta_{ij}$, $J_1 = \partial P_i/\partial\delta_j = -|V_i||V_j|B_{ij}$
  - since $|V_j|\approx 1$, off diagonal elements of $J_1 = \partial P_i/\partial\delta_j = -|V_i|B_{ij}$

FAST DECOUPLED POWER FLOW

- Consider the Newton-Raphson power flow equation
  - similarly, diagonal elements of J_4: $\partial Q_i/\partial|V_i| = -|V_i|B_{ii}$
  - off diagonal elements of J_4: $\partial Q_i/\partial|V_j| = -|V_i|B_{ij}$
  - therefore, ΔP and ΔQ has the following forms

$$\frac{\Delta P}{|V_i|} = -B^{'}\Delta\delta, \quad \frac{\Delta Q}{|V_i|} = -B^{''}\Delta|V|$$

  - B' and B" are the imaginary part of Y_{bus}
  - the updated Δδ and Δ|V| can be obtained from

$$\Delta\delta = -[B']^{-1}\frac{\Delta P}{|V|}, \quad \Delta|V| = -[B'']^{-1}\frac{\Delta Q}{|V|}$$

  - to calculate PQ bus, use simplified J_1 and J_4 to obtain solution
  - to calculate PV bus, J_4 can be further eliminated, only J_1 is used to obtain solution

FAST DECOUPLED POWER FLOW

- Comparison between fast decouple power flow solution and Newton Raphson power flow solution
  - fast decoupled solution requires more iterations than Newton Raphson solution
  - fast decoupled solution requires less time per iteration
  - since decoupled solution needs less time for iteration, the overall computation time may be less than using the Newton Raphson method
  - fast decoupled solution often used in fast computation of power flow, for example, contingency analysis or on-line control of power flow
  - *see Ex. 6.12*