Answers To Questions

1) a) Apply fault and clear fault, drop circuit.
 b) Apply fault; just remove row, column of faulted bus in \(Y'\), the un-reduced \(Y\)-bus with load shunts to get \(Y_2\), then reduce it.

c) Clear fault, Drop circuit:
\[
Y_2 = Y' - Y_{pp} \quad \text{where} \quad Y_{pp} \quad \text{has}\quad y_{pp} \quad \text{in elements} \quad p\quad p, \quad \text{pp and} \quad y_{pp}, \quad \text{and} \quad -y_{pp} \quad \text{in elements} \quad p\quad q, \quad q\quad p.
\]

Then for both \(Y_2\) and \(Y_3\), you need to reduce to get \(Y_2, Y_3\)

2) a) \(N = \#\) of generators
 b) \(P_{ei} = E_i^2 G_{ii} + \sum_{j=1}^{N} E_i E_j Y_{ij} \cos(\theta_{ij} - S_i + S_j)\)

\[= E_i^2 G_{ii} + \sum_{j=1}^{N} E_i E_j[\sin(\theta_i - S_i) + G_{ij} \cos(\theta_i - S_i)]\]

3) a) You can just use the internal bus voltage.
 b) For any particular generator, you only need the voltage (mag + angles) of the buses to which it is connected. These will only be other internal generator buses, since we are using \(Y_2\) and \(Y_3\), the reduced Y-bus matrices.

c) Step 3 is just an Euler integration step.
 b) We need to recompute \(P_{ei}\) before taking another step to obtain the \(P_{ei}(x'(t))\), which represents the powers at the predicted states (angles), which will be used to get the state derivatives at the predicted states for computing the corrected states \(x^c\) in step 5.

c) The argument of \(fi\) is a vector because \(fi\) contains \(P_{ei}\), which, in general, is a vector at all network angles.
There are 2N equations in step 3 because there are 2 state variables, \( S \) and \( W \), for each generator, and there are \( N \) generators.

\[
\dot{\omega}_i(t) = \frac{\text{wre}}{2H_i} \left[ P_{mi} - P_{ei} (S(t-T)) \right] + \omega_i(t-T)
\]

\[
\dot{S}_i(t) = S_i(t-T) + \omega_i(t-T) T
\]

4) This step makes the integration scheme the predictor-corrector (modified Euler).

\[
\frac{\text{step} \ 4}{\text{Step} \ 4}
\]

\[
\dot{\omega}_i^c(t) = \frac{1}{2} \left[ \dot{\omega}_i(t-T) + \dot{\omega}_i^c(t) \right]
\]

\[
\dot{\omega}_i(t-T) = \frac{\text{wre}}{2H_i} \left[ P_{mi} - P_{ei} (S(t-T)) \right]
\]

\[
\dot{\omega}_i^c(t) = \frac{\text{wre}}{2H_i} \left[ P_{mi} - P_{ei} (S^c(t)) \right]
\]

\[
\dot{S}_i^c(t) = \frac{1}{2} \left[ \dot{S}_i(t-T) + S_i^c(t) \right]
\]

Step 5

\[
\omega_i^c(t) = \omega_i(t-T) + \dot{\omega}_i^c T
\]

\[
S_i^c(t) = S_i(t-T) + \dot{S}_i^c T
\]

An improvement could be made by computing \( \omega_i^c \), then \( \omega_i^c \), then use

\[
\dot{S}_i^c(t) = \frac{1}{2} \left[ \omega_i(c(t-T) + \omega_i^c(t) \right]
\]

We use most recent update here.
The internal voltages and rotor angles for each generator are computed using the generated active and reactive powers, voltage magnitudes, and angles as:

\[
I_1 = \frac{(0.7164 + j0.2685)}{1.0400\angle 0^\circ} = 0.6888 + j0.2582
\]

\[
E_1 \angle \delta_1 = (j0.0608)(0.6888 - j0.2582) + 1.0400\angle 0^\circ = 1.0565\angle 2.2718^\circ
\]

\[
I_2 = \frac{(1.6300 + j0.0669)}{1.0253\angle 9.2715^\circ} = 1.5795 - j0.1918
\]

\[
E_2 \angle \delta_2 = (j0.1198)(1.5795 + j0.1918) + 1.0253\angle 9.2715^\circ = 1.0505\angle 19.7162^\circ
\]

\[
I_3 = \frac{(0.8500 - j0.1080)}{1.0254\angle 4.6587^\circ} = 0.8177 - j0.1723
\]

\[
E_3 \angle \delta_3 = (j0.1813)(0.8177 + j0.1723) + 1.0254\angle 4.6587^\circ = 1.0174\angle 13.1535^\circ
\]

The next step is to convert the loads to equivalent impedances:

\[
G_5 = \frac{(0.90 - j0.30)}{1.0128^2} = 0.8773 - j0.2924
\]

\[
G_7 = \frac{(1.00 - j0.35)}{1.0162^2} = 0.9684 - j0.3389
\]

\[
G_9 = \frac{(1.25 - j0.50)}{0.9958^2} = 1.2605 - j0.5042
\]

These values are added to the diagonal of the original admittance matrix.

\[
\begin{bmatrix}
17.3611 \angle -90.00^\circ & 0 & 0 & 0 & 17.3611 \angle 90.00^\circ & 0 & 0 & 0 & 0 \\
0 & 16.0000 \angle -90.00^\circ & 0 & 0 & 0 & 0 & 0 & 16.0000 \angle 90.00^\circ & 0 \\
0 & 0 & 17.3611 \angle 90.00^\circ & 0 & 0 & 0 & 0 & 17.3611 \angle -90.00^\circ & 0 \\
17.3611 \angle -90.00^\circ & 0 & 0 & 0 & 16.0000 \angle -90.00^\circ & 0 & 0 & 0 & 16.0000 \angle 90.00^\circ \\
0 & 0 & 0 & 39.4478 \angle -85.19^\circ & 10.6886 \angle 100.47^\circ & 0 & 0 & 0 & 11.6841 \angle -96.71^\circ \\
0 & 0 & 0 & 0 & 16.0000 \angle -90.00^\circ & 5.7334 \angle 102.92^\circ & 32.5481 \angle -85.67^\circ & 9.8522 \angle 96.73^\circ & 0 \\
0 & 0 & 0 & 0 & 0 & 16.0000 \angle 90.00^\circ & 5.7334 \angle -102.92^\circ & 32.5481 \angle 85.67^\circ & 9.8522 \angle -96.73^\circ \\
0 & 0 & 0 & 0 & 0 & 0 & 16.0000 \angle 90.00^\circ & 11.6841 \angle -96.71^\circ & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.0920 \angle 101.24^\circ & 17.5252 \angle -81.62^\circ
\end{bmatrix}
\]
The reduced admittance matrices can now be computed as outlined in steps 4 and 5 above. The pre-fault admittance matrix is

\[ Y_{\text{red-pre-fault}} = \begin{bmatrix} 0.8453 - j2.9881 & 0.2870 + j1.5131 & 0.2095 + j1.2257 \\ 0.2870 + j1.5131 & 0.4199 - j2.7238 & 0.2132 + j1.0880 \\ 0.2095 + j1.2257 & 0.2132 + j1.0880 & 0.2769 - j2.3681 \end{bmatrix} \]

The fault-on matrix is found similarly, except that the \( Y_{mm} \) is altered to reflect the fault on bus 8. The solid three-phase fault is modeled by shorting the bus to ground. In the admittance matrix, the row and column corresponding to bus 8 are removed. The lines between bus 8 and adjacent buses are now connected to ground; thus, they will still appear in the original admittance diagonals. The column of \( Y_{nm} \) and the row of \( Y_{mn} \) corresponding to bus 8 must also be removed. The matrix \( Y_{nn} \) remains unchanged. The fault-on reduced admittance matrix is

\[ Y_{\text{red-fault-on}} = \begin{bmatrix} 0.6567 - j3.8159 & 0 & 0.0701 + j0.6306 \\ 0 & 0 - j5.4855 & 0 \\ 0.0701 + j0.6306 & 0 & 0.1740 - j2.7959 \end{bmatrix} \]

The post-fault reduced admittance matrix is computed in much the same way, except that line 8-9 is removed from \( Y_{mm} \). The elements of \( Y_{mm} \) are updated to reflect the removal of the line:

\[ Y_{mm}(8, 8) = Y_{mm}(8, 8) + Y_{mm}(8, 9) \]
\[ Y_{mm}(8, 9) = Y_{mm}(9, 9) + Y_{mm}(8, 9) \]
\[ Y_{mm}(8, 9) = 0 \]
\[ Y_{mm}(9, 8) = 0 \]

Note that the diagonals must be updated before the off-diagonals are zeroed out. The post-fault reduced admittance is then computed:

\[ Y_{\text{red-post-fault}} = \begin{bmatrix} 1.1811 - j2.2285 & 0.1375 + j0.7265 & 0.1909 + j1.0795 \\ 0.1375 + j0.7265 & 0.3885 - j1.9525 & 0.1987 + j1.2294 \\ 0.1909 + j1.0795 & 0.1987 + j1.2294 & 0.2727 - j2.3423 \end{bmatrix} \]

These admittance matrices are then ready to be substituted into the transient stability equations at the appropriate time in the simulation.

Applying the trapezoidal algorithm to the transient stability equations yields the following system of equations:

\[ \delta_1(n + 1) = \delta_1(n) + \frac{h}{2} [\omega_1(n + 1) - \omega_s + \omega_1(n) - \omega_t] \quad (5.143) \]
\[ \omega_1(n + 1) = \omega_1(n) + \frac{h}{2} [f_1(n + 1) + f_1(n)] \quad (5.144) \]
\[ \delta_2(n + 1) = \delta_2(n) + \frac{h}{2} [\omega_2(n + 1) - \omega_s + \omega_2(n) - \omega_t] \quad (5.145) \]
\begin{equation}
\omega_2(n+1) = \omega_2(n) + \frac{h}{2} \left[ f_2(n+1) + f_2(n) \right]
\end{equation}

\begin{equation}
\delta_3(n+1) = \delta_3(n) + \frac{h}{2} \left[ \omega_3(n+1) - \omega_s + \omega_3(n) - \omega_s \right]
\end{equation}

\begin{equation}
\omega_3(n+1) = \omega_3(n) + \frac{h}{2} \left[ f_3(n+1) + f_3(n) \right]
\end{equation}

where

\begin{equation}
f_i(n+1) = \frac{1}{M_i} \left( P_{mi} - E_i^2 G_{ii} - E_i \sum_{j \neq i} E_j (B_{ij} \sin \delta_{ij}(n+1) + G_{ij} \cos \delta_{ij}(n+1)) \right)
\end{equation}

Since the transient stability equations are nonlinear and the trapezoidal method is an implicit method, they must be solved iteratively using the Newton-Raphson method at each time point. The iterative equations are

\begin{equation}
\begin{bmatrix}
\delta_1(n+1) \vphantom{\omega_1(n+1)} \\
\omega_1(n+1) \\
\delta_2(n+1) \\
\omega_2(n+1) \\
\delta_3(n+1) \\
\omega_3(n+1)
\end{bmatrix}
= 
\begin{bmatrix}
\delta_1(n+1)^{k+1} - \delta_1(n+1)^k \\
\omega_1(n+1)^{k+1} - \omega_1(n+1)^k \\
\delta_2(n+1)^{k+1} - \delta_2(n+1)^k \\
\omega_2(n+1)^{k+1} - \omega_2(n+1)^k \\
\delta_3(n+1)^{k+1} - \delta_3(n+1)^k \\
\omega_3(n+1)^{k+1} - \omega_3(n+1)^k
\end{bmatrix}
\end{equation}

- \begin{bmatrix}
\delta_1(n+1) \\
\omega_1(n+1) \\
\delta_2(n+1) \\
\omega_2(n+1) \\
\delta_3(n+1) \\
\omega_3(n+1)
\end{bmatrix}
- \frac{h}{2}
\begin{bmatrix}
\delta_1(n) \\
\omega_1(n) \\
\delta_2(n) \\
\omega_2(n) \\
\delta_3(n) \\
\omega_3(n)
\end{bmatrix}
\begin{bmatrix}
\omega_1(n+1) + \omega_1(n) - 2\omega_s \\
f_1(n+1) + f_1(n) \\
\omega_2(n+1) + \omega_2(n) - 2\omega_s \\
f_2(n+1) + f_2(n) \\
\omega_3(n+1) + \omega_3(n) - 2\omega_s \\
f_3(n+1) + f_3(n)
\end{bmatrix}

where

\begin{equation}
[J] =
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
\frac{\partial f_1}{\partial \delta_1} & 0 & \frac{\partial f_1}{\partial \delta_2} & 0 & \frac{\partial f_1}{\partial \delta_3} & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
\frac{\partial f_2}{\partial \delta_1} & 0 & \frac{\partial f_2}{\partial \delta_2} & 0 & \frac{\partial f_2}{\partial \delta_3} & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\frac{\partial f_3}{\partial \delta_1} & 0 & \frac{\partial f_3}{\partial \delta_2} & 0 & \frac{\partial f_3}{\partial \delta_3} & 0
\end{bmatrix}
\end{equation}

Note that LU factorization must be employed to solve the discretized equations. These equations are iterated at each time point until convergence of the Newton-Raphson algorithm. The fault-on and post-fault matrices are substituted in at the appropriate times in the integration. The simulation results are shown in Figures 5.18 and 5.19 for the rotor angles and angular frequencies, respectively. From the waveforms shown in these figures, it can be concluded that the system remains stable since the waveforms do not diverge during the simulation interval.
FIGURE 5.18
Rotor angle response for Example 5.9

FIGURE 5.19
Angular frequency response for Example 5.9