EE 581 Power Systems

Admittance Matrix: Direct and Iterative solutions, Sensitivity

Overview and HW # 9

- Chapter 6.1
- Chapter 6.2
- Chapter 6.3

- Homework: 6.23 (4^{th} edition)/6.22 (5^{th} edition)) and Special Problem 3

Overview and HW # 9

- Homework: 6.23/6.22
- Use Newton-Raphson to find a solution to
- $\begin{bmatrix} e^{x_1 x_2} \\ \cos(x_1 + x_2) \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.5 \end{bmatrix}$ where x_1 and x_2 are in radians.
- A) start with $x_1(0) = 1.0$ and $x_2(0) = 0.5$ and continue until $\left|\frac{x_k(i+1) - x_k(i)}{x_k(i)}\right| < \varepsilon$ with $\varepsilon = 0.005$.
- B)Show that this method diverges if $x_1(0) = 1.0$ and $x_2(0) = 2.0$

Overview and HW # 9

- Homework: Special problem 3
- Given $A = \begin{bmatrix} 10 & 20 & 33 \\ 41.7 & 26 & 61 \\ 77 & 82 & 91 \end{bmatrix} \quad b = \begin{bmatrix} 901 \\ 542 \\ 817 \end{bmatrix}$
- In Matlab, using the tic and toc commands:
- A) compare the time used for x=inv(A)*b, x=A\b, lu decomposition (use [L,U]=lu(A) command),.
- B) using the Jacobi method : $\mathrm{D} \cdot x(i+1) = (D - A) \cdot x(i) + b$, start at k=1 iterations and go to k=10. Comment on the time and the convergence of the method.

Overview and HW # 9

- Homework: Special problem 3
- A) check: $x = \begin{bmatrix} -41.5773 \\ 14.4272 \\ 31.1584 \end{bmatrix}$

Chapter 6.1: Gaussian Elimination

- Development/review:
- Given a matrix of size (n x n)
- Can be solved using the following elementary operations:
  - I. Interchange two rows
  - II. Multiply a row by a nonzero real number
  - III. Replace a row by its sum with a multiple of another row

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & & \vdots & \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Chapter 6.1: Gaussian Elimination

- Solution for row two:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ 0 & \left(A_{22} - \frac{A_{21}}{A_{11}} A_{12}\right) & \cdots & \left(A_{2N} - \frac{A_{21}}{A_{11}} A_{1N}\right) \\ 0 & \left(A_{32} - \frac{A_{31}}{A_{11}} A_{12}\right) & \cdots & \left(A_{3N} - \frac{A_{31}}{A_{11}} A_{1N}\right) \\ \vdots & \vdots & & \vdots \\ 0 & \left(A_{N2} - \frac{A_{N1}}{A_{11}} A_{12}\right) & \cdots & \left(A_{NN} - \frac{A_{N1}}{A_{11}} A_{1N}\right) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 - \frac{A_{21}}{A_{11}} y_1 \\ y_3 - \frac{A_{31}}{A_{11}} y_1 \\ \vdots \\ y_N - \frac{A_{N1}}{A_{11}} y_1 \end{bmatrix}$$

Chapter 6.1: Gaussian Elimination

- Final format (row echelon form):
- This can be further simplified to where $A_{ii} = 1,\ i = 1 \ldots N$
- This is reduced row echelon form and can be computed in MATLAB using rref(A)

$$\begin{bmatrix} A_{11} & A_{12}\ldots & & A_{1N} \\ 0 & A_{22}\ldots & & A_{2N} \\ \vdots & & & \\ 0 & 0\ldots & A_{N-1,N-1} & A_{N-1,N} \\ 0 & 0\ldots 0 & & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}$$

Chapter 6.1: Gaussian Elimination

- Example 6.1
- Solve: $\begin{bmatrix} 10 & 5 \\ 2 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$ in matrix notation $\begin{bmatrix} row\ 1 \\ row\ 2 \end{bmatrix} \vec{x} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
- Solution
- $\begin{bmatrix} r_1 \\ r_2 - \left(\frac{2}{10}\right) r_1 \end{bmatrix} \vec{x} = \begin{bmatrix} y_1 \\ y_2 - \left(\frac{2}{10}\right) y_1 \end{bmatrix}$
- $\begin{bmatrix} 10 & 5 \\ 2 - 2 & 9 - \left(\frac{2}{10}\right) 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 - \left(\frac{2}{10}\right) 6 \end{bmatrix}$
- $\begin{bmatrix} 10 & 5 \\ 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 1.8 \end{bmatrix}$

Chapter 6.1: Gaussian Elimination

- Example 6.1 cont.
- $\begin{bmatrix} 10 & 5 \\ 0 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 1.8 \end{bmatrix}$
- Using back substitution:
- $x_2 = \frac{y_2}{A_{22}} = \frac{1.8}{8} = 0.225$
- $x_1 = \frac{y_1 - A_{12}x_2}{A_{11}} = \frac{6-(5)(0.225)}{10} = 0.4875$

Chapter 6.1: LU Decomposition

- Similarly LU is the matrix form of Gaussian Elimination, which also allows for the use of sparse matrices.

- A=LU such that: (3x3 case) :
- $$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$
- This allows the solution of $Ax = y$ in the form:
- 1) $Lb = y$
- 2)$Ux = b$

Chapter 6.1: LU Decomposition

- Example 6.1 using LU:
- A=LU
- using matlab [L,U] = lu(A) we obtain
- $\begin{bmatrix} 10 & 5 \\ 2 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0.2 & 1 \end{bmatrix} \begin{bmatrix} 10 & 5 \\ 0 & 9 \end{bmatrix}$
- $b = L\backslash \mathrm{y} = \begin{bmatrix} 6 \\ 1.8 \end{bmatrix}$
- $x = U\backslash \mathrm{b} = \begin{bmatrix} 0.4875 \\ 0.225 \end{bmatrix} \longleftarrow$ which is the same as in example 6.1

Chapter 6.2: Jacobi and Gauss-Seidel

- Iterative methods involve approximations of linear systems over k iterations that continue until a stopping criteria is met
- Things to consider:
  1. Will the iteration method converge?
  2. What is the convergence rate (number of iterations)?
  3. When using a computer, what are the storage and time constraints for the method being used? (usually storage is the more pertinent issue)
- A measure of the convergence can be expressed as a tolerance level:
- $\left|\frac{x_k(i+1)-x_k(i)}{x_k(i)}\right| < \varepsilon \; for \; k = 1,2 \ldots N$ and $Ax = y, \;\; where \; x \; is \; (N \; x1)$
- ε is typically chosen to be a very small number, relative to the size of the elements in the A and y matrices

Chapter 6.2: Jacobi and Gauss-Seidel

- Considering the equation for the k^{th} element we can solve for x:
- $y_k = A_{k1}x_1 + A_{k2}x_{k2} + \cdots + A_{kk}x_k + \cdots + A_{kn}x_N$
- $x_k = \frac{1}{A_{kk}}[y_k - (A_{k1}x_1 + \cdots + A_{kn}x_N)]$

  $= \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m - \sum_{m=k+1}^{N} A_{km}x_m]$
- For each k^{th} element, this equation is iterated i times, such that we can write the Jacobi method as:
- $x_k(i+1) = \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m(i) - \sum_{m=k+1}^{N} A_{km}x_m\,(i)]$

  $where\ x_k(i+1) is\ the\ newest\ iteration$

Chapter 6.2: Jacobi and Gauss-Seidel

- Similarly for Gauss-Seidel, considering the equation for the k^{th} element we can solve for x to obtain:
- $x_k = \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m - \sum_{m=k+1}^{N} A_{km}x_m]$ ← Same as Jacobi!
- The key difference is with Gauss-Seidel, for $m < k$, the x values used are the updated ones, where as in Jacobi the original starting x values.
- For Gauss-Seidel:
- $x_k(i+1) = \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m(i+1) - \sum_{m=k+1}^{N} A_{km}x_m\,(i)]$

- Note for both Gauss-Seidel and Jacobi, $x_k(0) = 0$ typically

Chapter 6.2: Jacobi and Gauss-Seidel

- Both of these methods can be expressed in matrix form:
- $x(i+1) = Mx(i) + D^{-1}y \quad where\ M = D^{-1}(D-A)$

- Jacobi:
- $x_k(i+1) = \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m(i) - \sum_{m=k+1}^{N} A_{km}x_m(i)]$
- $D = diagonal(A)$

- Gauss-Seidel:
- $x_k(i+1) = \frac{1}{A_{kk}}[y_k - \sum_{m=1}^{k-1} A_{km}x_m(i+1) - \sum_{m=k+1}^{N} A_{km}x_m(i)]$
- $D = lower\_triangular(A)$

Chapter 6.2: Jacobi and Gauss-Seidel

- Example 6.3 (6.1 using Jacobi) Ax=y $\begin{bmatrix} 10 & 5 \\ 2 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$
- Equation 1: $x_1(i+1) = \frac{1}{A_{11}}[y_1 - A_{12}x_2(i)] = \frac{1}{10}[6 - 5x_2(i)]$
- Equation 2: $x_2(i+1) = \frac{1}{A_{22}}[y_2 - A_{21}x_1(i)] = \frac{1}{9}[3 - 2x_1(i)]$

Original x values

- Matrix form:
- $D^{-1} = \begin{bmatrix} 10 & 0 \\ 0 & 9 \end{bmatrix}^{-1} = \begin{bmatrix} 0.1 & 0 \\ 0 & \frac{1}{9} \end{bmatrix}$ $M = D^{-1}(D - A) = \begin{bmatrix} 0 & -0.5 \\ -\frac{2}{9} & 0 \end{bmatrix}$
- $x(i+1) = Mx(i) + D^{-1}y$
- $= \begin{bmatrix} 0 & -0.5 \\ -\frac{2}{9} & 0 \end{bmatrix} \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix} + \begin{bmatrix} 0.1 & 0 \\ 0 & \frac{1}{9} \end{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix}$

Chapter 6.2: Jacobi and Gauss-Seidel

- Example 6.3 (6.1 using Jacobi) cont.
- Iterations with goal of $\varepsilon < 10^{-4}$

JACOBI

i	0	1	2	3	4	5	6	7	8	9	10
$x_1(i)$	0	0.60000	0.43334	0.50000	0.48148	0.48889	0.48683	0.48766	0.48743	0.48752	0.48749
$x_2(i)$	0	0.33333	0.20000	0.23704	0.22222	0.22634	0.22469	0.22515	0.22496	0.22502	0.22500

As shown, the Jacobi method converges to the unique solution obtained in Example 6.1. The convergence criterion is satisfied at the 10th iteration, since

$$\left|\frac{x_1(10) - x_1(9)}{x_1(9)}\right| = \left|\frac{0.48749 - 0.48752}{0.48749}\right| = 6.2 \times 10^{-5} < \varepsilon$$

and

$$\left|\frac{x_2(10) - x_2(9)}{x_2(9)}\right| = \left|\frac{0.22500 - 0.22502}{0.22502}\right| = 8.9 \times 10^{-5} < \varepsilon \quad \blacksquare$$

Chapter 6.2: Jacobi and Gauss-Seidel

- Example 6.4 (6.1 using Gauss-Seidel)

SOLUTION From (6.2.9),

$$k = 1 \qquad x_1(i+1) = \frac{1}{A_{11}}[y_1 - A_{12}x_2(i)] = \frac{1}{10}[6 - 5x_2(i)]$$

$$k = 2 \qquad x_2(i+1) = \frac{1}{A_{22}}[y_2 - A_{21}x_1(i+1)] = \frac{1}{9}[3 - 2x_1(i+1)]$$

Using this equation for $x_1(i+1)$, $x_2(i+1)$ can also be written as

$$x_2(i+1) = \frac{1}{9}\left\{3 - \frac{2}{10}[6 - 5x_2(i)]\right\}$$

Alternatively, in matrix format, using (6.2.10), (6.2.6), and (6.2.7):

$$\mathbf{D}^{-1} = \left[\begin{array}{c|c} 10 & 0 \\ \hline 2 & 9 \end{array}\right]^{-1} = \left[\begin{array}{c|c} \frac{1}{10} & 0 \\ \hline -\frac{2}{90} & \frac{1}{9} \end{array}\right]$$

Chapter 6.2: Jacobi and Gauss-Seidel

- Example 6.4 (6.1 using Gauss-Seidel)

$$\mathbf{M} = \begin{bmatrix} \frac{1}{10} & 0 \\ -\frac{2}{90} & \frac{1}{9} \end{bmatrix} \begin{bmatrix} 0 & -5 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{9} \end{bmatrix}$$

$$\begin{bmatrix} x_1(i+1) \\ x_2(i+1) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{9} \end{bmatrix} \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix} + \begin{bmatrix} \frac{1}{10} & 0 \\ -\frac{2}{90} & \frac{1}{9} \end{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

These two formulations are identical. Starting with $x_1(0) = x_2(0) = 0$, the solution is given in the following table:

GAUSS–SEIDEL

i	0	1	2	3	4	5	6
$x_1(i)$	0	0.60000	0.50000	0.48889	0.48765	0.48752	0.48750
$x_2(i)$	0	0.20000	0.22222	0.22469	0.22497	0.22500	0.22500

For this example, Gauss–Seidel converges in 6 iterations, compared to 10 iterations with Jacobi. ■

Chapter 6.2: Jacobi and Gauss-Seidel

- Divergence and Method failure:
  - No iterative solution
  - A_{kk} diagonal element is << the other diagonals
  - Fails if $A_{kk} = 0$ due to $x_k = \frac{1}{0}\left[y_k - \sum_{m=1}^{k-1} A_{km}x_m - \sum_{m=k+1}^{N} A_{km}x_m\right]$, which is undefined!

- Further issues can be caused by:
  - Setting the tolerance level ε too small (causing an infinite loop)
  - Setting ε too large and obtaining a wrong solution set
  - For N dimension size being very large, not enough memory
- Jacobi uses $N^2 + 3N$ storage space, while GS uses $N^2 + 2N$ due to the updated x values

Chapter 6.2: Jacobi and Gauss-Seidel

- Divergence via no iterative solution:
- Example 6.5
- Using Gauss-Seidel to solve $\begin{bmatrix} 5 & 10 \\ 9 & 2 \end{bmatrix} \vec{x} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$
- Equation 1: $x_1(i+1) = \frac{1}{5}[6 - 10x_2(i)]$
- Equation 2: $x_2(i+1) = \frac{1}{2}[3 - 9x_1(i+1)]$

Successive calculations of x_1 and x_2 are shown in the following table:

GAUSS–SEIDEL

i	0	1	2	3	4	5
$x_1(i)$	0	1.2	9	79.2	711	6397
$x_2(i)$	0	−3.9	−39	−354.9	−3198	−28786

- Using $x = A\backslash y$ we find the exact soln. is $\begin{bmatrix} 0.225 \\ 0.4875 \end{bmatrix}$ $\therefore$ the method diverges!

Chapter 6.3: Newton-Raphson

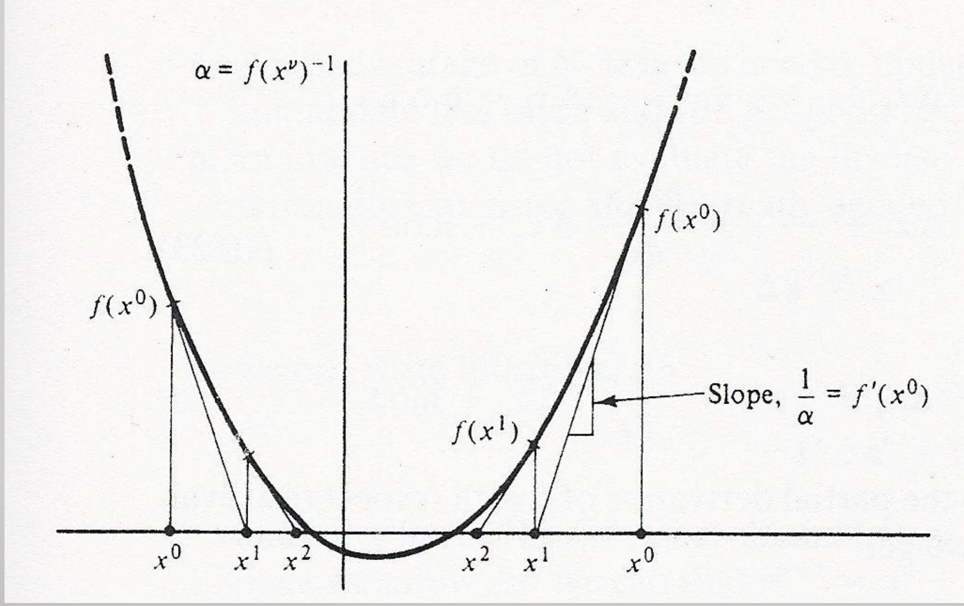
- Given $Ax = y,\ let\ f(x) = Ax \quad s.t.$
- $f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_N(x) \end{bmatrix} = y$
- To develop the iterative soln.
- $y - f(x) = 0$
- $Dx + f(x) = Dx + y \quad where\ \ D\ is\ (N \times N)$and $x\ is\ the\ soln.vector$
- $x = x + D^{-1}[y - f(x)]$
- $x(i+1) = x(i) + D^{-1}\{y - f[x(i)]\}$

Chapter 6.3: Newton-Raphson

- Convergence properties:
- $x(i+1) = x(i) + D^{-1}\{y - f[x(i)]\}$, ignoring the y term,
- $x(i+1) = x(i) - D^{-1}f[x(i)]$, with special case $D^{-1} = \alpha I \equiv \alpha$
- $x(i+1) = x(i) - \alpha f[x(i)]$
- $\left(\frac{1}{\alpha}\right)x(i+1) = \left(\frac{1}{\alpha}\right)x(i) - f[x(i)]$

Chapter 6.3: Newton-Raphson

- $\left(\frac{1}{\alpha}\right) x(i+1) = \left(\frac{1}{\alpha}\right) x(i) - f[x(i)]$
- Solving for $f(x) = 0$ (again ignoring y for inspection) and varying α, we find the graph this to be:



Here we see can map the slope as $\frac{1}{\alpha} = f'(x)$ $\therefore$ we can say

$$f'(x)x(i+1) = f'(x)x(i) - f[x(i)]$$

Chapter 6.3: Newton-Raphson

- $f'(x)x(i+1) = f'(x)x(i) - f[x(i)]$
- $x(i+1) = x(i) - f'(x)^{-1} \cdot f[x(i)]$, which can be rewritten as
- $x(i+1) = x(i) - \left[\frac{df}{dx}\right]^{-1} \cdot f[x(i)] \leftarrow$ so we are looking for something like this
- Returning to $x(i+1) = x(i) + D^{-1}\{y - f[x(i)]\}$
- To find D, y can be expanded using a Taylor series as:
- $y = f(x_i) + \frac{df}{dx}|_{x=x_i}(x - x_i) \ldots$ about the point x_i
- $y_1 = f(x_1) + \left[\frac{\partial f_1}{\partial x_1} + \frac{\partial f_1}{\partial x_2} + \cdots + \frac{\partial f_1}{\partial x_N}\right](x - x_1)$
- $y_2 = f(x_2) + \left[\frac{\partial f_2}{\partial x_1} + \frac{\partial f_2}{\partial x_2} + \cdots + \frac{\partial f_2}{\partial x_N}\right](x - x_2)$

Chapter 6.3: Newton-Raphson

- Solving $y_i = f(x_i) + \left[\frac{\partial f_i}{\partial x_1} + \frac{\partial f_i}{\partial x_2} + \cdots + \frac{\partial f_i}{\partial x_N}\right](x - x_i)$
- $J_i(x - x_i) = [y - f(x_i)]$ multiplying each side by J_i^{-1}
- $x = x_i + J_i^{-1}[y - f(x_i)]$
- Thus the Newton-Raphson method in matrix iteration format is:
- $x(i+1) = x(i) + J(i)^{-1}\{y - f[x(i)]\}$ where
- $J(i) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix}_{x=x(i)}$

Chapter 6.3: Newton-Raphson

- Example 6.7
- Solve $\begin{bmatrix} x_1 + x_2 \\ x_1 x_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 50 \end{bmatrix}$ with $x_0 = \begin{bmatrix} 4 \\ 9 \end{bmatrix}$
- $J(i) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ x_2(i) & x_1(i) \end{bmatrix} \quad \Rightarrow \quad J(i)^{-1} = \begin{bmatrix} x_1(i) & -1 \\ -x_2(i) & 1 \end{bmatrix}$
- $x(i+1) = x(i) + J(i)^{-1} \{y - f[x(i)]\}$
- $\begin{bmatrix} x_1(i+1) \\ x_2(i+1) \end{bmatrix} = \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix} + \begin{bmatrix} x_1(i) & -1 \\ -x_2(i) & 1 \end{bmatrix} \begin{bmatrix} 15 - (x_1 + x_2) \\ 50 - (x_1 x_2) \end{bmatrix}$

Chapter 6.3: Newton-Raphson

- Example 6.7 cont.
- Separating each equation and iterating for $\varepsilon < 10^{-4}$

$$x_1(i+1) = x_1(i) + \frac{x_1(i)[15 - x_1(i) - x_2(i)] - [50 - x_1(i)x_2(i)]}{x_1(i) - x_2(i)}$$

$$x_2(i+1) = x_2(i) + \frac{-x_2(i)[15 - x_1(i) - x_2(i)] + [50 - x_1(i)x_2(i)]}{x_1(i) - x_2(i)}$$

Successive calculations of these equations are shown in the following table:

NEWTON–RAPHSON

i	0	1	2	3	4
$x_1(i)$	4	5.20000	4.99130	4.99998	5.00000
$x_2(i)$	9	9.80000	10.00870	10.00002	10.00000

Newton–Raphson converges in four iterations for this example. ■