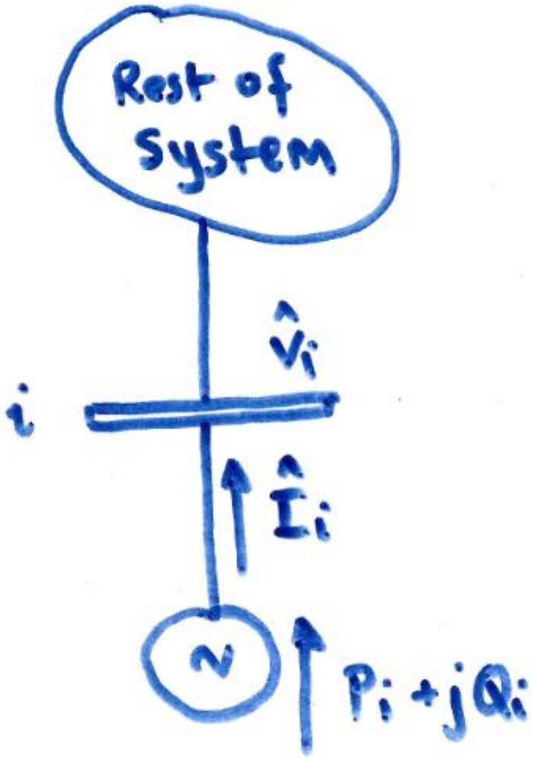
THE UNIVERSITY of ADELAIDE

Course:
ELEC ENG 3110 Electric Power Systems
ELEC ENG 7074 Power Systems PG
(Semester 2, 2021)

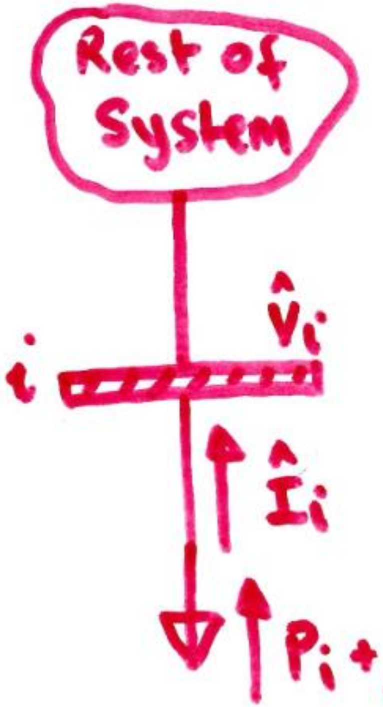
Powerflow Analysis (Part 2)

Lecturer and Co-ordinator: David Vowles
david.vowles@adelaide.edu.au

Nodal power and reactive power injections (1)



Generator Bus
(PV, VA)



Load Bus
(PQ)

$$P_i + jQ_i = -(P_{Li} + jQ_{Li})$$

Convert from load to generator convention.

$$P_i + jQ_i = \hat{V}_i \hat{I}_i^*$$

$$\hat{I}_i = \sum_{k=1}^{N} Y_{ik} \hat{V}_k$$

$$P_i + jQ_i = \hat{V}_i \left(\sum_{k=1}^{N} Y_{ik}^* \hat{V}_k^* \right)$$

Product ⇒ NON LINEAR

© The University of Adelaide

Nodal power and reactive power injections (2)

Nodal current injections unknown.

Non-linearly related to P, Q, V, θ at each node

$$\hat{I}_i = \sum_{k=1}^{n} (\underline{Y_{ik}} \hat{V}_k) = \sum_{k=1}^{n} (\underline{G_{ik} + jB_{ik}})(V_k e^{j\theta_k})$$

$$P_i + jQ_i = \hat{V}_i \underline{\hat{I}_i^*} = \hat{V}_i \underline{\left[\sum_{k=1}^{n} (G_{ik} - jB_{ik}) V_k e^{-j\theta_k}\right]}$$

$$P_i + jQ_i = V_i e^{j\theta_i} \sum_{k=1}^{n} (G_{ik} - jB_{ik}) V_k e^{-j\theta_k}$$

© The University of Adelaide

Nodal power and reactive power injections (3)

$$V_i e^{j\theta_i} \cdot V_k e^{-j\theta_k} = V_i V_k e^{j(\theta_i - \theta_k)}$$

$$= V_i V_k e^{j\theta_{ik}} \qquad , \ \theta_{ik} = \theta_i - \theta_k$$

$$= V_i V_k (\cos\theta_{ik} + j\sin\theta_{ik})$$

$$(G_{ik} - jB_{ik}) V_i V_k e^{j\theta_{ik}} = V_i V_k (G_{ik} - jB_{ik})(\cos\theta_{ik} + j\sin\theta_{ik})$$

$$= V_i V_k \left([G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik}] + j[-B_{ik}\cos\theta_{ik} + G_{ik}\sin\theta_{ik}] \right)$$

$$\boxed{\begin{aligned} P_i &= V_i \sum_{k=1}^{n} (G_{ik} V_k \cos\theta_{ik} + B_{ik} V_k \sin\theta_{ik}) \\ Q_i &= V_i \sum_{k=1}^{n} (G_{ik} V_k \sin\theta_{ik} - B_{ik} V_k \cos\theta_{ik}) \end{aligned}}$$

$$\boxed{\theta_{ik} = \theta_i - \theta_K}$$

Nodal power and reactive power injection equations

Nodal power and reactive power constraints (1)

For each node i for which P is specified we can write :

$$\boxed{P_{is}} - \underbrace{P_i(\overbrace{V_1, V_2 .. V_n}^{\underset{\sim}{V}}, \overbrace{\theta_1, \theta_2, \dots \theta_n}^{\underset{\sim}{\theta}})}_{\text{Nodal power injection equation for node } i} = 0$$

specified power injection

(function of all nodal voltage magnitudes and angles)

NOTE: A load PL + jQL specified in accordance with the load convention must be negated to become a power injection. That is, Pis = -PL and Qis = -QL.

$$\underset{\sim}{V} = [V_1, V_2, \dots V_n]^T \qquad \underset{\sim}{\theta} = [\theta_1, \theta_2, \dots \theta_n]^T$$

Nodal power and reactive power constraints (2)

Similarly for each node i for which Q is specified (i.e. PQ nodes) —

$$\underbrace{Q_{is}}_{\text{specified reactive power injection}} - \underbrace{Q_i(\underset{\sim}{V}, \underset{\sim}{\theta})}_{\text{nodal reactive-power injection equation for node } i} = 0$$

Ordering of network nodes for computational convenience (1)

For computational convenience we order the network nodes as follows:

$$\left.\begin{array}{lll} 1 & \text{VA bus} & -\; 1 \\ n & \text{PV buses} & -\; 2 \rightarrow n+1 \end{array}\right\} \text{ Voltages known}$$

$$\left.\begin{array}{lll} m & \text{PQ buses} & -\; n+2 \rightarrow n+1+m \end{array}\right\} \text{ unknown.}$$

The voltage vector is partitioned as

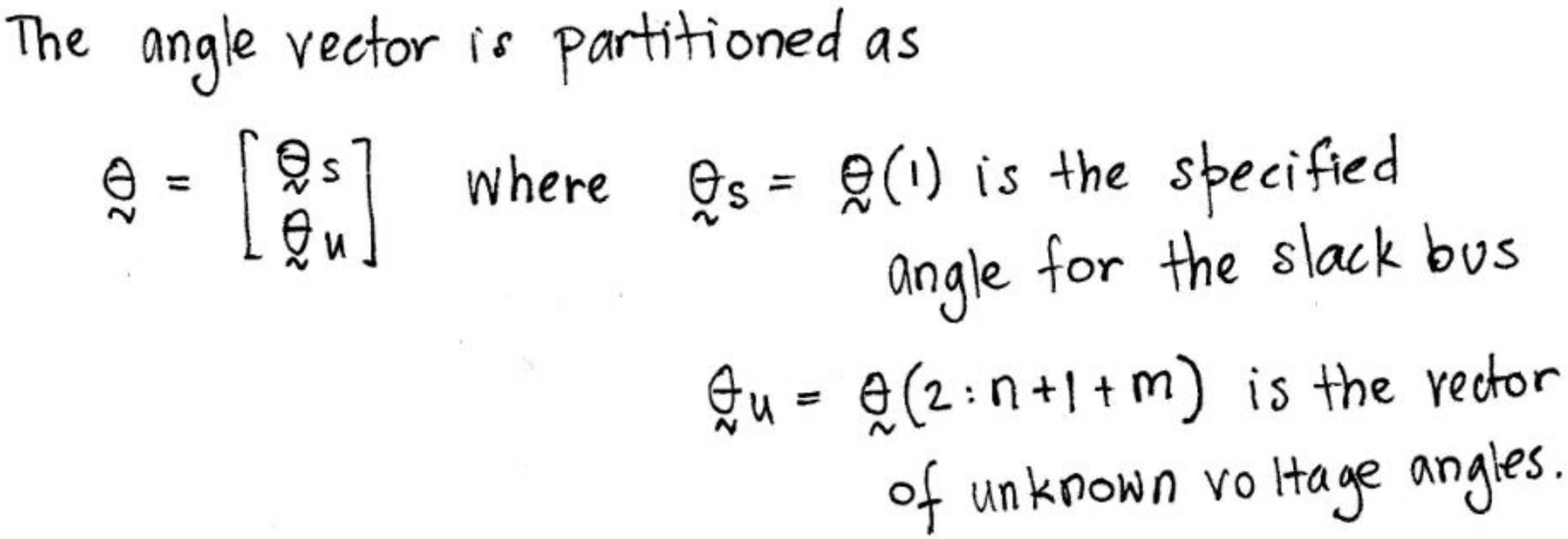
$$\underset{\sim}{V} = \begin{bmatrix} \underset{\sim}{V}_s \\ \underset{\sim}{V}_u \end{bmatrix}$$

where $\underset{\sim}{V}_s = \underset{\sim}{V}(1:n+1)$ is the vector of specified voltages

$\underset{\sim}{V}_u = \underset{\sim}{V}(n+2:n+1+m)$ is the vector of unknown voltages.

© The University of Adelaide

Ordering of network nodes for computational convenience (2)



The angle vector is partitioned as

$$\underset{\sim}{\theta} = \begin{bmatrix} \underset{\sim}{\theta}_s \\ \underset{\sim}{\theta}_u \end{bmatrix}$$

where $\underset{\sim}{\theta}_s = \underset{\sim}{\theta}(1)$ is the specified angle for the slack bus

$\underset{\sim}{\theta}_u = \underset{\sim}{\theta}(2:n+1+m)$ is the vector of unknown voltage angles.

© The University of Adelaide

We assemble the power equations as follows .

No power equation for VA (slack) bus

$$n+m \text{ equations} \begin{cases} P_{2s} - P_2(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \\ P_{3s} - P_3(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \\ \vdots \\ P_{rs} - P_r(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \end{cases} \quad , \quad r = n+1+m$$

n PV nodes
m PQ nodes

And the reactive power equations

$$m \text{ equations} \begin{cases} Q_{ws} - Q_w(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \quad , \quad w = n+2 \\ Q_{(w+1)s} - Q_{(w+1)}(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \\ \vdots \\ Q_{rs} - Q_r(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0 \end{cases}$$

No reactive power equations for VA or PV buses

Equation / Variable audit

$n+m$	power equations
m	reactive power equations
$n+2m$	equations.

m	unknown voltages
$n+m$	unknown angles
$n+2m$	unknown variables

Equal number of equations and unknown variables

Vectorized formulation of the power-flow equations

$$\underset{\sim}{P}_s - \underset{\sim}{P}(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0$$

$$\underset{\sim}{Q}_s - \underset{\sim}{Q}(\underset{\sim}{V}_s, \underset{\sim}{V}_u, \underset{\sim}{\theta}_s, \underset{\sim}{\theta}_u) = 0$$

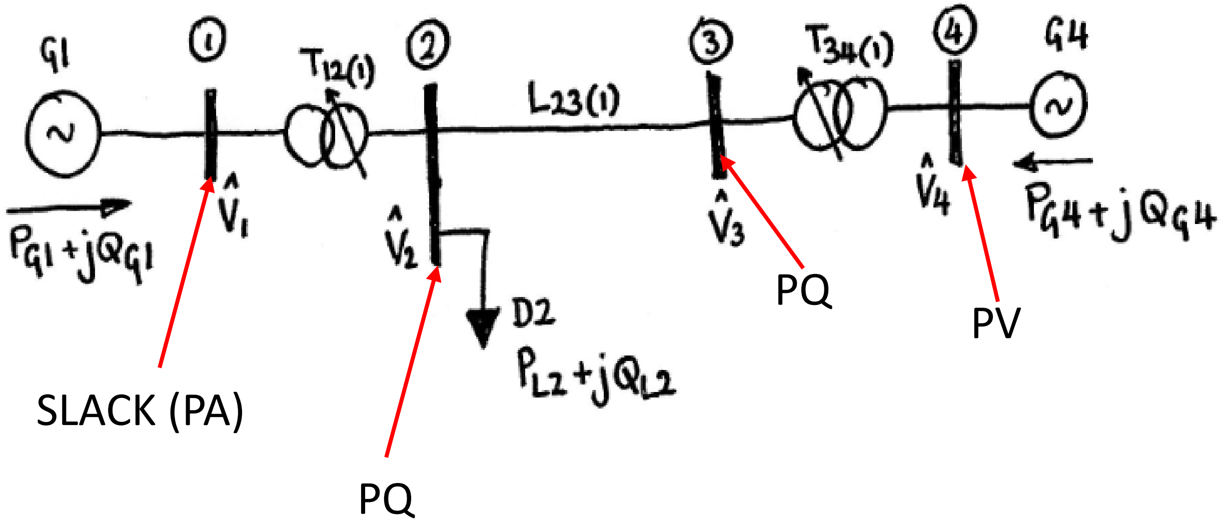
We use iterative methods to solve this system of non-linear equations.

NOTE: The power and reactive power output (P, Q) of the slack bus and the reactive power outputs (Q) of the PV buses are computed once the above power-flow equations are solved using the equations for P & Q on slide 11.

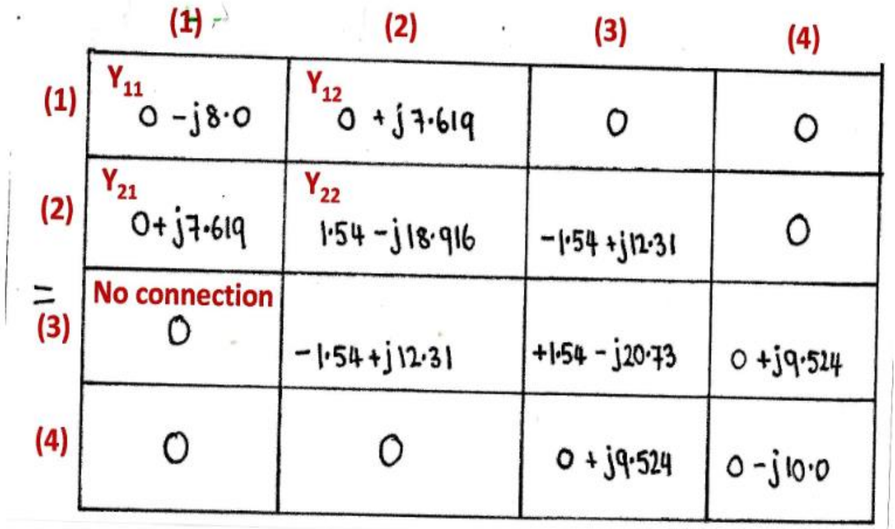
Overview of the power-flow solution

- Wide range of solution methods.
- Two complementary methods are:
  - Gauss-Seidel Method.
    - Converges slowly especially for large systems
    - Tolerant of poor initial guesses
    - Useful in obtaining an initial estimate near actual solution for other fast converging algorithms
  - Newton-Raphson (NR) based methods
    - Quadratic convergence rate providing initial estimate is close to actual solution
    - Variants of the NR method based on properties of transmission networks
      - Such as decoupling between voltage and angle (Fast-Decoupled)
      - Avoid updating the Jacobian matrix at each iteration
- We will review the basic NR method
- Sophisticated power-flow tools provide an extensive range of facilities:
  - Automatic area interchange controls
  - Automatic transformer tap-change controls
  - Optimal power flow solver (economic dispatch, loss minimization)
  - PV and QV analysis
  - Automated contingency analysis
  - Sophisticated data base facilities and graphical displays

Power flow equations of simple example (1)



$[Y]=[G+jB]$ =



	(1)	(2)	(3)	(4)
(1)	Y_{11} 0 − j8.0	Y_{12} 0 + j7.619	0	0
(2)	Y_{21} 0+j7.619	Y_{22} 1.54 − j18.916	−1.54 +j12.31	0
(3)	No connection 0	−1.54+j12.31	+1.54 − j20.73	0 +j9.524
(4)	0	0	0 + j9.524	0 − j10.0

$$P_i = V_i \sum_{k=1}^{n} \left(G_{ik} V_k \cos\theta_{ik} + B_{ik} V_k \sin\theta_{ik} \right)$$

$$Q_i = V_i \sum_{k=1}^{n} \left(G_{ik} V_k \sin\theta_{ik} - B_{ik} V_k \cos\theta_{ik} \right)$$

$$\theta_{ik} = \theta_i - \theta_k$$

Nodal power and reactive power injection equations

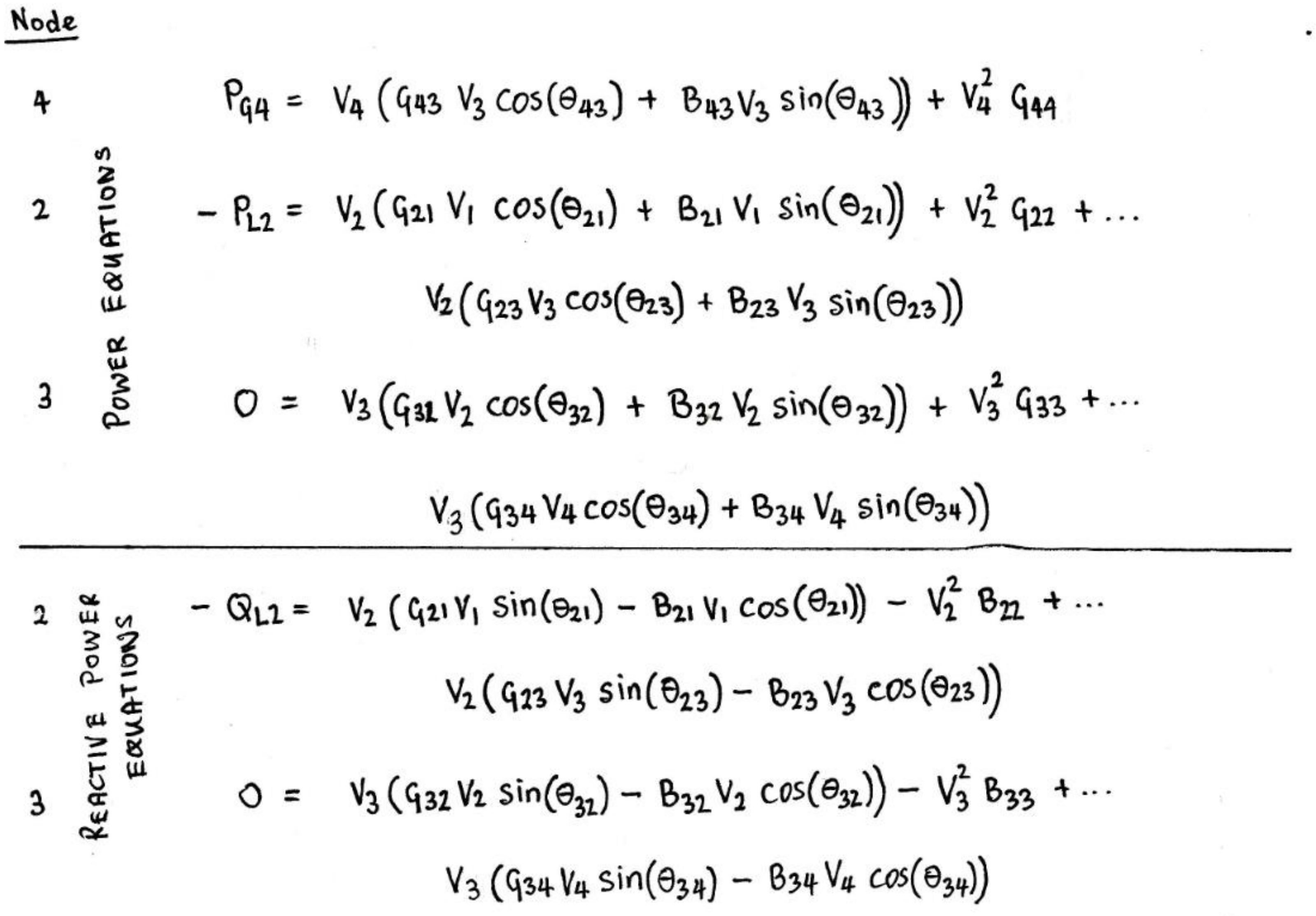
© The University of Adelaide

Power flow equations of simple example (2)

Write power equations for PV and PQ buses only. Omit power equation for slack bus. Thus, three power equations.

Write reactive power equations for PQ buses only. Thus, two reactive power equations.

Highly non-linear equations



Node

POWER EQUATIONS

4 $$P_{G4} = V_4\left(G_{43}V_3\cos(\theta_{43}) + B_{43}V_3\sin(\theta_{43})\right) + V_4^2 G_{44}$$

2 $$-P_{L2} = V_2\left(G_{21}V_1\cos(\theta_{21}) + B_{21}V_1\sin(\theta_{21})\right) + V_2^2 G_{22} + \dots$$
$$V_2\left(G_{23}V_3\cos(\theta_{23}) + B_{23}V_3\sin(\theta_{23})\right)$$

3 $$0 = V_3\left(G_{32}V_2\cos(\theta_{32}) + B_{32}V_2\sin(\theta_{32})\right) + V_3^2 G_{33} + \dots$$
$$V_3\left(G_{34}V_4\cos(\theta_{34}) + B_{34}V_4\sin(\theta_{34})\right)$$

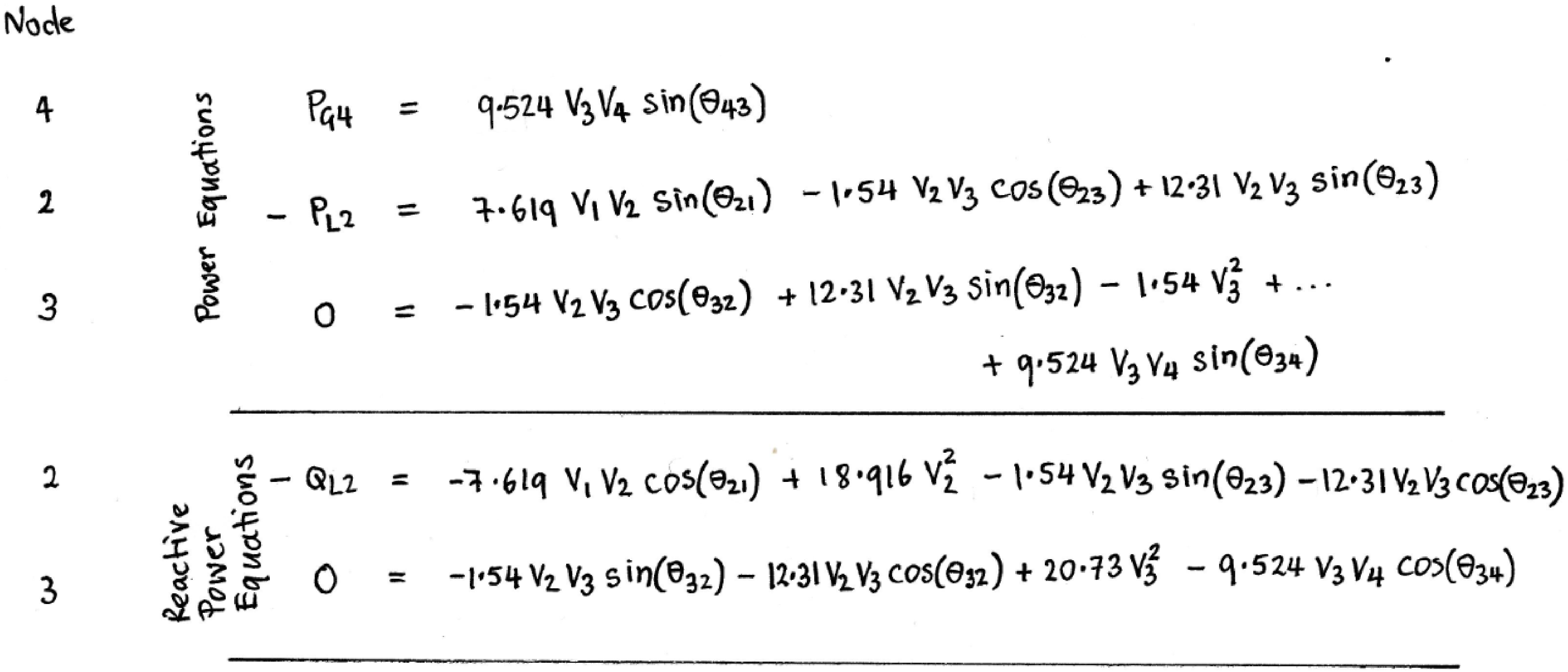
REACTIVE POWER EQUATIONS

2 $$-Q_{L2} = V_2\left(G_{21}V_1\sin(\theta_{21}) - B_{21}V_1\cos(\theta_{21})\right) - V_2^2 B_{22} + \dots$$
$$V_2\left(G_{23}V_3\sin(\theta_{23}) - B_{23}V_3\cos(\theta_{23})\right)$$

3 $$0 = V_3\left(G_{32}V_2\sin(\theta_{32}) - B_{32}V_2\cos(\theta_{32})\right) - V_3^2 B_{33} + \dots$$
$$V_3\left(G_{34}V_4\sin(\theta_{34}) - B_{34}V_4\cos(\theta_{34})\right)$$

Power flow equations of simple example (3)

Substitute numerical values for conductance (Gik) and susceptance (Bik) terms from the network admittance matrix.



Node		
4	Power Equations	$P_{G4} = 9{\cdot}524\,V_3V_4\sin(\theta_{43})$
2		$-P_{L2} = 7{\cdot}619\,V_1V_2\sin(\theta_{21}) - 1{\cdot}54\,V_2V_3\cos(\theta_{23}) + 12{\cdot}31\,V_2V_3\sin(\theta_{23})$
3		$0 = -1{\cdot}54\,V_2V_3\cos(\theta_{32}) + 12{\cdot}31\,V_2V_3\sin(\theta_{32}) - 1{\cdot}54\,V_3^2 + \dots + 9{\cdot}524\,V_3V_4\sin(\theta_{34})$
2	Reactive Power Equations	$-Q_{L2} = -7{\cdot}619\,V_1V_2\cos(\theta_{21}) + 18{\cdot}916\,V_2^2 - 1{\cdot}54\,V_2V_3\sin(\theta_{23}) - 12{\cdot}31\,V_2V_3\cos(\theta_{23})$
3		$0 = -1{\cdot}54\,V_2V_3\sin(\theta_{32}) - 12{\cdot}31\,V_2V_3\cos(\theta_{32}) + 20{\cdot}73\,V_3^2 - 9{\cdot}524\,V_3V_4\cos(\theta_{34})$

© The University of Adelaide

Introduction to Newton-Raphson Method – Single Variable (1)

Solve $f(x) = y$ for x

Let x^0 be initial estimate, Δx^* be correction so

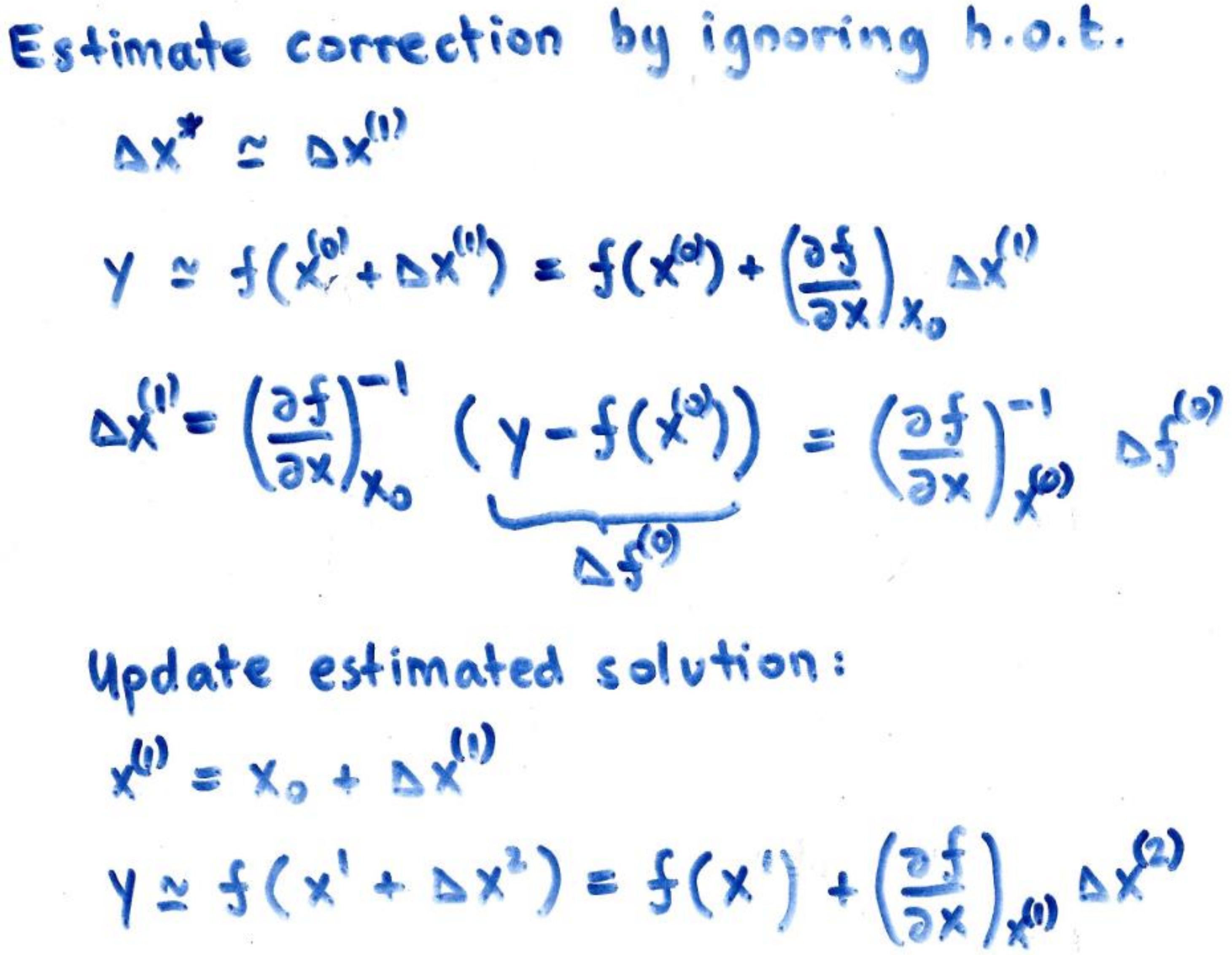
$$x = x^0 + \Delta x^*$$ is the exact solution.

$$f(x) = f(x^0 + \Delta x^*) = y$$

Taylors series expansion

$$y = f(x^0 + \Delta x^*) = f(x^0) + \left(\frac{\partial f}{\partial x}\right)_{x_0} \Delta x^* + \underbrace{\frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_{x_0} (\Delta x^*)^2 + \ldots}_{\text{h.o.t.}}$$

Introduction to Newton-Raphson Method – Single Variable (2)



Estimate correction by ignoring h.o.t.

$$\Delta x^{*} \simeq \Delta x^{(1)}$$

$$y \simeq f(x^{(0)} + \Delta x^{(1)}) = f(x^{(0)}) + \left(\frac{\partial f}{\partial x}\right)_{x_0} \Delta x^{(1)}$$

$$\Delta x^{(1)} = \left(\frac{\partial f}{\partial x}\right)_{x_0}^{-1} \underbrace{\left(y - f(x^{(0)})\right)}_{\Delta f^{(0)}} = \left(\frac{\partial f}{\partial x}\right)_{x^{(0)}}^{-1} \Delta f^{(0)}$$

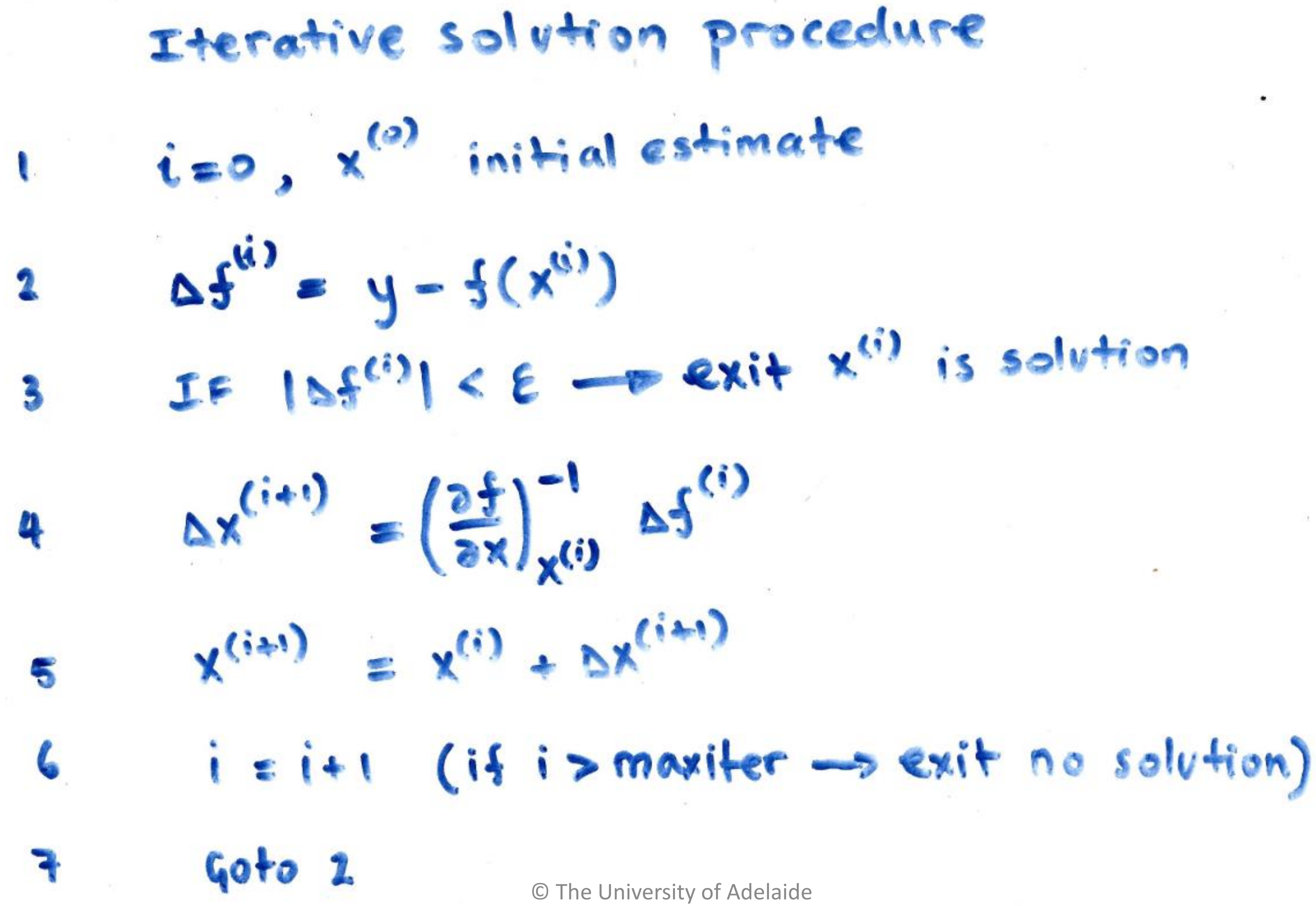
Update estimated solution:

$$x^{(1)} = x_0 + \Delta x^{(1)}$$

$$y \simeq f(x^{1} + \Delta x^{2}) = f(x^{1}) + \left(\frac{\partial f}{\partial x}\right)_{x^{(1)}} \Delta x^{(2)}$$

© The University of Adelaide

Introduction to Newton-Raphson Method – Single Variable (3)



Iterative solution procedure

```
1   $i = 0$ , $x^{(0)}$ initial estimate
2   $\Delta f^{(i)} = y - f(x^{(i)})$
3   IF $|\Delta f^{(i)}| < \varepsilon$ → exit $x^{(i)}$ is solution
4   $\Delta x^{(i+1)} = \left(\frac{\partial f}{\partial x}\right)^{-1}_{x^{(i)}} \Delta f^{(i)}$
5   $x^{(i+1)} = x^{(i)} + \Delta x^{(i+1)}$
6   $i = i + 1$ (if $i >$ maxiter → exit no solution)
7   Goto 2
```

© The University of Adelaide

Introduction to Newton-Raphson Method – Single Variable (4)

Example

$f(x) = x^2 = 5$, $\frac{\partial f}{\partial x} = 2x$, $x^{(0)} = 2$

i	$\Delta f^{(i)} = 5-(x^{(i)})^2$	$\left(\frac{\partial f}{\partial x}\right)^{-1}_{x^{(i)}}$	$\Delta x^{(i+1)}$	$x^{(i+1)}$
0	$5-4=1$	$\frac{1}{2\times2} = 0.25$	1×0.25 $= 0.25$	$2+0.25$ $= 2.25$
1	$5-2.25^2$ $= -0.0625$	$\frac{1}{2\times2.25} = 0.2222$	-0.0139	$2.25-0.0139$ $= 2.2361$
2	$5-2.2361^2$ $= -1.929\times10^{-4}$	$\frac{1}{2\times2.2361} = 0.2236$	-4.3133×10^{-5}	$\boxed{2.236068}$
3	$5-2.236068^2$ $= -1.86\times10^{-9} \longrightarrow$ solved			$x = \sqrt{5} = 2.236068...$

Summary of the Newton-Raphson Method – Multi Variable (1)

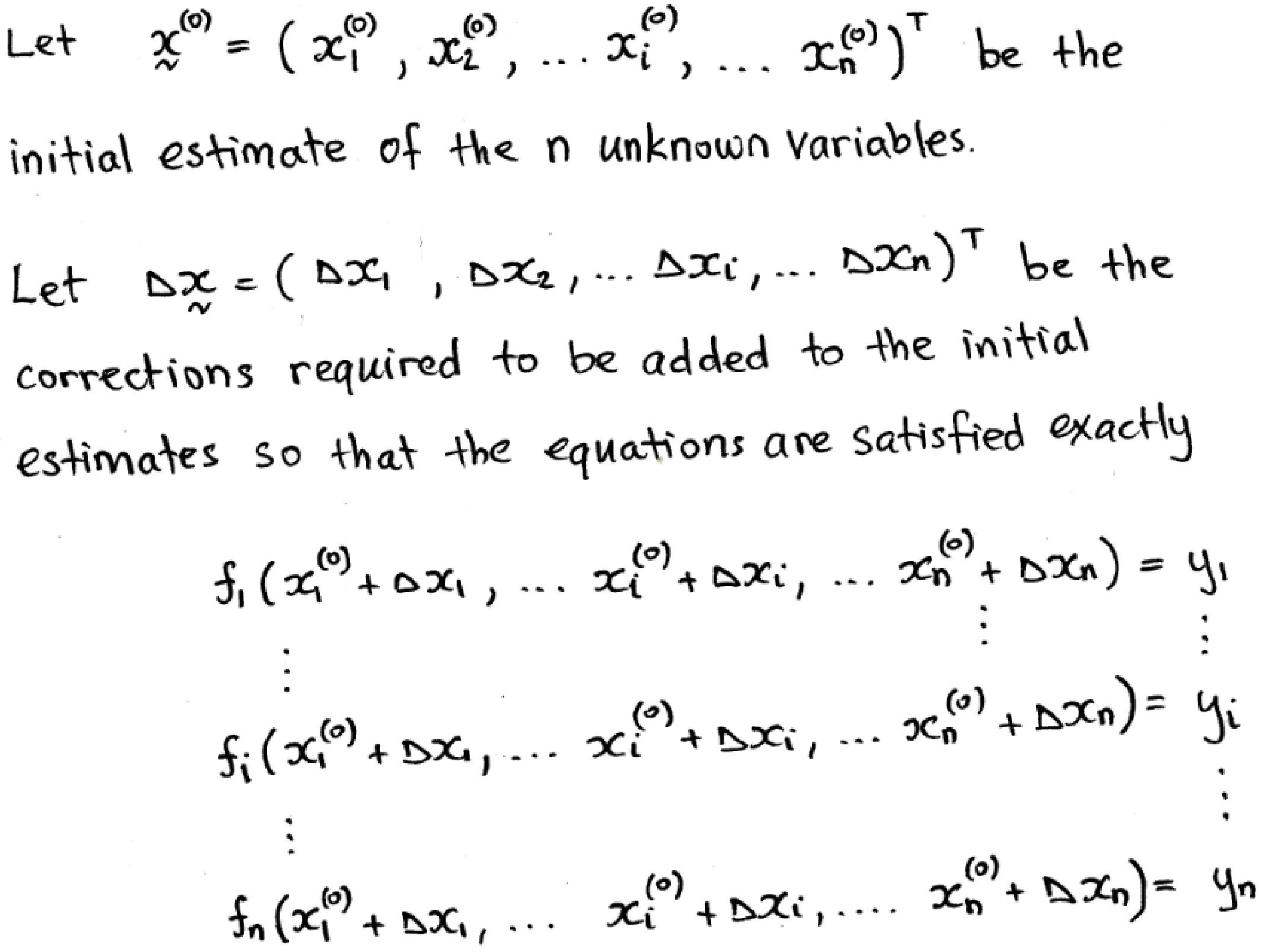
- Consider a system on *n* nonlinear equations in *n* unknowns

$$
\begin{aligned}
f_1(x_1, x_2, \dots x_i, \dots x_n) &= y_1 \\
f_2(x_1, x_2, \dots x_i, \dots x_n) &= y_2 \\
&\vdots \\
f_i(x_1, x_2, \dots x_i, \dots x_n) &= y_i \\
&\vdots \\
f_n(x_1, x_2, \dots x_i, \dots x_n) &= y_n
\end{aligned}
$$

$$\underset{\sim}{x} = (x_1, x_2, \dots x_i, \dots x_n)^T \quad \text{[unknown variables]}$$

$$\underset{\sim}{y} = (y_1, y_2, \dots y_i, \dots y_n)^T$$

Summary of the Newton-Raphson Method – Multi Variable (2)



Let $\underset{\sim}{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots x_i^{(0)}, \ldots x_n^{(0)})^T$ be the initial estimate of the n unknown variables.

Let $\Delta\underset{\sim}{x} = (\Delta x_1 , \Delta x_2 , \ldots \Delta x_i , \ldots \Delta x_n)^T$ be the corrections required to be added to the initial estimates so that the equations are satisfied exactly

$$
\begin{array}{l}
f_1 (x_1^{(0)} + \Delta x_1 , \ldots x_i^{(0)} + \Delta x_i , \ldots x_n^{(0)} + \Delta x_n) = y_1 \\
\vdots \\
f_i (x_1^{(0)} + \Delta x_1 , \ldots x_i^{(0)} + \Delta x_i , \ldots x_n^{(0)} + \Delta x_n) = y_i \\
\vdots \\
f_n (x_1^{(0)} + \Delta x_1 , \ldots x_i^{(0)} + \Delta x_i , \ldots x_n^{(0)} + \Delta x_n) = y_n
\end{array}
$$

Summary of the Newton-Raphson Method – Multi Variable (3)

Consider the i^{th} equation

$$f_i\,(x_1^{o}+\Delta x_1,\ x_2^{o}+\Delta x_2,\ \dots\ x_i^{(o)}+\Delta x_i,\ \dots\ x_n^{(o)}+\Delta x_n) = y_i$$

Expand using Taylors Theorem :

$$f_i(x_1^{o},\ x_2^{o},\ \dots\ x_i^{(o)},\ \dots\ x_n^{(o)})\ +$$

$$\left(\frac{\partial f_i}{\partial x_1}\right)_0 \Delta x_1 + \left(\frac{\partial f_i}{\partial x_2}\right)_0 \Delta x_2 + \dots \left(\frac{\partial f_i}{\partial x_i}\right)_0 \Delta x_i + \dots \left(\frac{\partial f_i}{\partial x_n}\right)_0 \Delta x_n + \text{h.o.t.}$$

$$= y_i$$

© The University of Adelaide

Summary of the Newton-Raphson Method – Multi Variable (4)

Express in vectorized form:

$$y_i = f_i(\underset{\sim}{x}^{(0)}) + \underbrace{\left[\left(\frac{\partial f_i}{\partial x_1}\right)_0 \left(\frac{\partial f_i}{\partial x_2}\right)_0 \cdots \left(\frac{\partial f_i}{\partial x_i}\right)_0 \cdots \left(\frac{\partial f_i}{\partial x_n}\right)_0\right]}_{\left(\frac{\partial f_i}{\partial \underset{\sim}{x}}\right)_0} \underbrace{\begin{bmatrix}\Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_i \\ \vdots \\ \Delta x_n\end{bmatrix}}_{\Delta \underset{\sim}{x}} + \text{h.o.t.}$$

$$\left(y_i - f_i(\underset{\sim}{x}^{(0)})\right) = \left(\frac{\partial f_i}{\partial \underset{\sim}{x}}\right)_0 \Delta \underset{\sim}{x} + \text{h.o.t.}$$

Summary of the Newton-Raphson Method – Multi Variable (5)

1xn row vector

$$\underbrace{\begin{bmatrix} y_1 - f_1(\underset{\sim}{x}^{(0)}) \\ y_2 - f_2(\underset{\sim}{x}^{(0)}) \\ \vdots \\ y_i - f_i(\underset{\sim}{x}^{(0)}) \\ \vdots \\ y_n - f_n(\underset{\sim}{x}^{(0)}) \end{bmatrix}}_{\Delta \underset{\sim}{f} = \underset{\sim}{y} - \underset{\sim}{f}(\underset{\sim}{x}^{(0)})} = \underbrace{\begin{bmatrix} \left(\frac{\partial f_1}{\partial \underset{\sim}{x}}\right)_0 \\ \left(\frac{\partial f_2}{\partial \underset{\sim}{x}}\right)_0 \\ \vdots \\ \left(\frac{\partial f_i}{\partial \underset{\sim}{x}}\right)_0 \\ \vdots \\ \left(\frac{\partial f_n}{\partial \underset{\sim}{x}}\right)_0 \end{bmatrix}}_{J = \left[\frac{\partial \underset{\sim}{f}}{\partial \underset{\sim}{x}}\right]} \Delta \underset{\sim}{x}$$

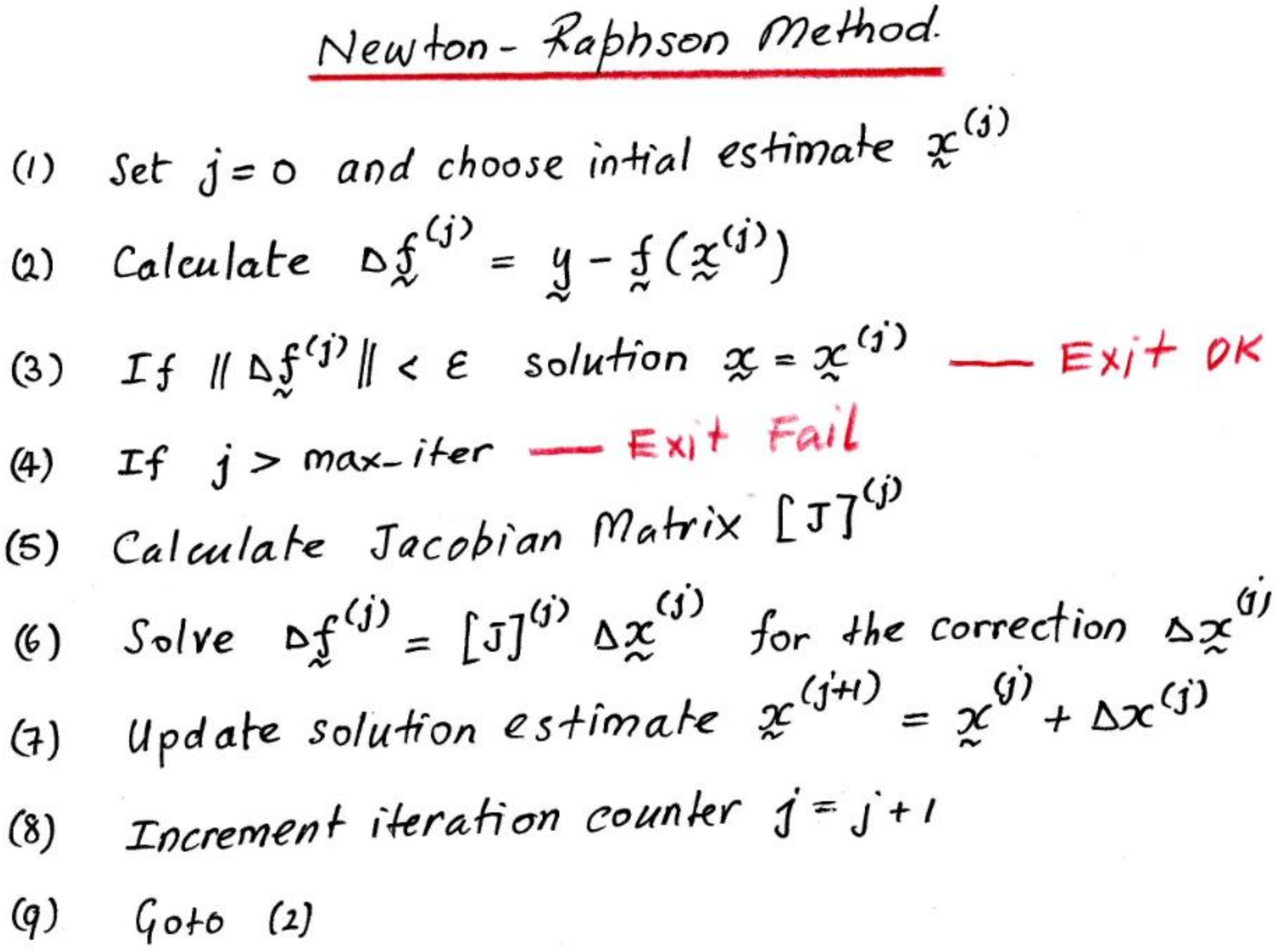
1xn row vector

$J = \left[\frac{\partial \underset{\sim}{f}}{\partial \underset{\sim}{x}}\right]$ ← Jacobian Matrix

$$\boxed{\Delta \underset{\sim}{f} = [J] . \Delta \underset{\sim}{x}}$$

© The University of Adelaide

Newton-Raphson Algorithm (General)



Newton-Raphson Method.

```
(1)  Set $j = 0$ and choose intial estimate $\underset{\sim}{x}^{(j)}$
(2)  Calculate $\Delta \underset{\sim}{f}^{(j)} = \underset{\sim}{y} - \underset{\sim}{f}(\underset{\sim}{x}^{(j)})$
(3)  If $\| \Delta \underset{\sim}{f}^{(j)} \| < \varepsilon$ solution $\underset{\sim}{x} = \underset{\sim}{x}^{(j)}$ — Exit OK
(4)  If $j >$ max_iter — Exit Fail
(5)  Calculate Jacobian Matrix $[J]^{(j)}$
(6)  Solve $\Delta \underset{\sim}{f}^{(j)} = [J]^{(j)} \Delta \underset{\sim}{x}^{(j)}$ for the correction $\Delta \underset{\sim}{x}^{(j)}$
(7)  Update solution estimate $\underset{\sim}{x}^{(j+1)} = \underset{\sim}{x}^{(j)} + \Delta x^{(j)}$
(8)  Increment iteration counter $j = j + 1$
(9)  Goto (2)
```

© The University of Adelaide