LOAD FLOW

A. J. Conejo

Univ. Castilla – La Mancha

2011

The load flow problem

0. References
1. Introduction
2. Problem formulation
   - Two-bus case
   - Matrix $\overline{Y}_{BUS}$
   - General equations
   - Bus classification
   - Variable types and limits

The load flow problem

3. The Gauss-Seidel solution technique
   - Introduction
   - Algorithm initialization
   - PQ Buses
   - PV Buses
   - Stopping criterion

The load flow problem

4. The Newton-Raphson solution technique

    Introduction

    General fomulation

    Load flow case

    Jacobian matrix

    Solution outline

The load flow problem

5. Fast decoupled AC load flow
6. Adjustment of bounds
7. DC load flow
8. Comparison of load flow solution methods

The load flow problem
References

1. A. Gómez Expósito, A. J. Conejo, C. Cañizares. “Electric Energy Systems. Analysis and Operation”. CRC Press, Boca Raton, Florida, 2008.

2. A. R. Bergen, V. Vittal. “Power Systems Analysis”. Second Edition. Prentice Hall, Upper Saddle River, New Jersey, 1999.

1. Introduction

Introduction

- A snapshot of the system

- Most used tool in steady state power system analysis

- Knowning the demand and/or generation of power in each bus, find out:
  - buses voltages
  - load flow in lines and transformers

Introduction

- The problem is described throught a non-lineal system of equations
- Need of iterative solution techniques
- Solution technique: accuracy vs. computing time

Introduction

- Applications:
  1. On-line analyses
     - State estimation
     - Security
     - Economic analyses

Introduction

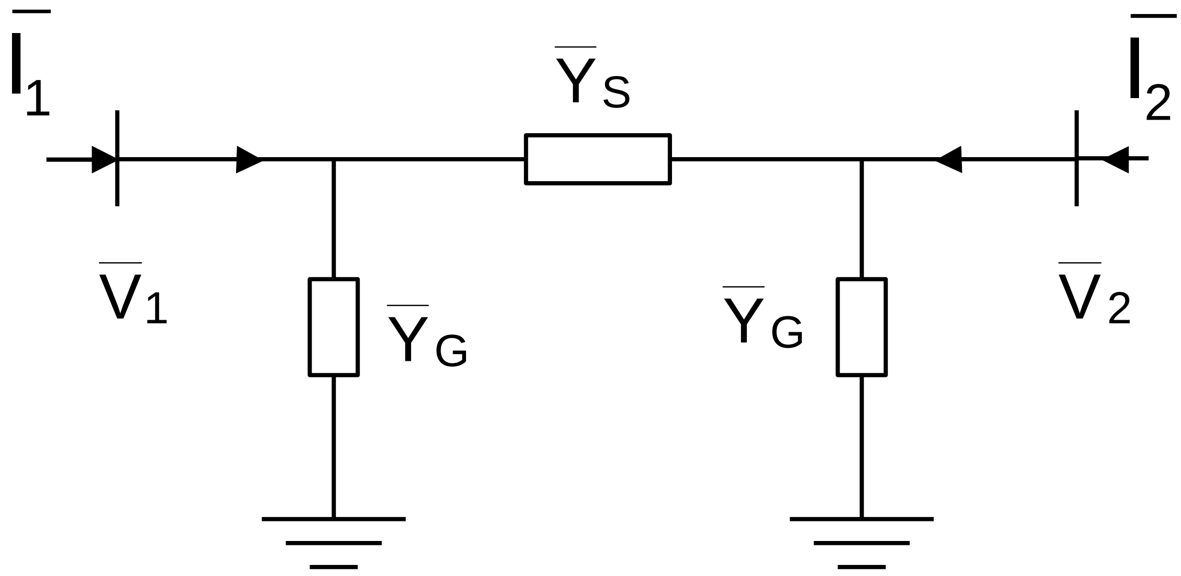
2. Off-line analyses

- Operation analyses
- Plannig analyses
  - ✓Network expansion planning
  - ✓Power exchange planning
  - ✓Security and adecuacy analyses
    - - Faults
    - - Stability

2. Problem formulation

Problem formulation
Two-bus case

We want to find out the relationship between $S_i = P_i + jQ_i$ and $\overline{V}_i = V_i e^{j\delta_i}$ in all buses of the power system



Transmission line π model

Problem formulation
Two-bus case

Using Kirchhoff laws:

$$\bar{I}_1 = \bar{V}_1\,\bar{Y}_G + (\bar{V}_1 - \bar{V}_2)\,\bar{Y}_S$$

$$\bar{I}_2 = \bar{V}_2\,\bar{Y}_G + (\bar{V}_2 - \bar{V}_1)\,\bar{Y}_S$$

Matrix notation:

$$\begin{bmatrix} \bar{I}_1 \\ \bar{I}_2 \end{bmatrix} = \begin{bmatrix} \bar{Y}_G + \bar{Y}_S & -\bar{Y}_S \\ -\bar{Y}_S & \bar{Y}_G + \bar{Y}_S \end{bmatrix} \times \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \end{bmatrix} = \begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12} \\ \bar{Y}_{21} & \bar{Y}_{22} \end{bmatrix} \times \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \end{bmatrix}$$

$$\left[\bar{I}_{BUS}\right] = \left[\bar{Y}_{BUS}\right] \times \left[\bar{V}_{BUS}\right]$$

Problem formulation
Two-bus case

- Complex power injected in each bus:

$$\overline{S}_1 = \overline{S}_{G1} - \overline{S}_{D1} = P_1 + jQ_1 = \overline{V}_1 \times \overline{I}_1^*$$

$$\overline{S}_2 = \overline{S}_{G2} - \overline{S}_{D2} = P_2 + jQ_2 = \overline{V}_2 \times \overline{I}_2^*$$

$$\overline{S}_1 = \overline{V}_1 \times \overline{I}_1^* = \overline{V}_1 \times (Y_{11}^* \ \overline{V}_1^* + Y_{12}^* \ \overline{V}_2^*)$$

$$\overline{S}_2 = \overline{V}_2 \times \overline{I}_2^* = \overline{V}_2 \times (Y_{21}^* \ \overline{V}_1^* + Y_{22}^* \ \overline{V}_2^*)$$

Problem formulation
Two-bus case continuation

Notation:

$$\overline{Y}_{ik} = Y_{ik}\ e^{j\theta_{ik}}$$

$$\overline{V}_i = V_i\ e^{j\delta_i}$$

Replacing:

$$P_1 + jQ_1 = V_1 \sum_{k=1}^{2} Y_{1k}\ V_k\ e^{j(\delta_1 - \delta_k - \theta_{1k})}$$

$$P_2 + jQ_2 = V_2 \sum_{k=1}^{2} Y_{2k}\ V_k\ e^{j(\delta_2 - \delta_k - \theta_{2k})}$$

Problem formulation
Two-bus case continuation

Therefore, the no lineal equations for the 2 buses network are:

$$P_i = V_i \times \sum_{k=1}^{2} Y_{ik}\ V_k \cos(\delta_i - \delta_k - \theta_{ik})$$

$$Q_i = V_i \times \sum_{k=1}^{2} Y_{ik}\ V_k \sin(\delta_i - \delta_k - \theta_{ik})$$

$$i = 1,2$$

Problem formulation
Matrix Y_{bus}

- Two bus case

$$\overline{Y}_{BUS} = \begin{bmatrix} \overline{Y}_{11} & \overline{Y}_{12} \\ \overline{Y}_{21} & \overline{Y}_{22} \end{bmatrix} = \begin{bmatrix} \overline{Y}_G + \overline{Y}_S & -\overline{Y}_S \\ -\overline{Y}_S & \overline{Y}_G + \overline{Y}_S \end{bmatrix}$$

General building rules
Matrix Y_{bus}

1. Self admittance of node i, $\overline{Y}_{ii}$,equals the algebraic sum of all the admittances connected to node i

2. Mutual admittance between nodes i and k, $\overline{Y}_{ik}$,equals the negative of the sum of all admittances connecting nodes i and k

3. $\overline{Y}_{ik} = \overline{Y}_{ki}$

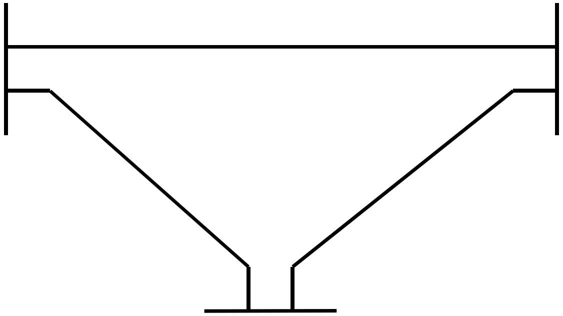
Problem formulation
Matrix Y_{bus}

- Caracteristics of $\overline{Y}_{BUS}$

1. $\overline{Y}_{BUS}$ is symmetric

2. $\overline{Y}_{BUS}$ is very sparse
   (>90% for more than 100 buses)

Y_{bus} example



Shunt element $\bar{Y}_G = j0.01$ (two per line)

Series element $\bar{Z}_S = j0.1$

$$\bar{Y}_{BUS} = \begin{bmatrix} -j19.98 & j10 & j10 \\ j10 & -j19.98 & j10 \\ j10 & j10 & -j19.98 \end{bmatrix}$$

Problem formulation
General equations

- 2n equations (static load flow equations)

$$P_i = P_{Gi} - P_{Di} = V_i \times \sum_{k=1}^{n} Y_{ik} V_k \cos(\delta_i - \delta_k - \theta_{ik}) \qquad [1]$$

$$Q_i = Q_{Gi} - Q_{Di} = V_i \times \sum_{k=1}^{n} Y_{ik} V_k \sin(\delta_i - \delta_k - \theta_{ik}) \qquad [2]$$

$i = 1, \ldots, n$

n – bus system

Problem formulation
General equations

Polar representation for voltages and rectangular for admittances

$$P_i = V_i \times \sum_{k=1}^{n} V_k \left(G_{ik} \cos\delta_{ik} + B_{ik} \operatorname{sen}\delta_{ik}\right) \qquad [1]$$

$$Q_i = V_i \times \sum_{k=1}^{n} V_k \left(G_{ik} \operatorname{sen}\delta_{ik} - B_{ik} \cos\delta_{ik}\right) \qquad [2]$$

where

$$\overline{Y}_{ik} = G_{ik} + jB_{ik}$$

$$\delta_{ik} = \delta_i - \delta_k$$

Problem formulation
General equations

- 4n variables

$$V_i, \delta_i, P_i, Q_i \qquad ; \qquad i = 1, \ldots, n$$

- If 2n variables are specified, the other 2n are determined by equations [1] and [2]

Problem formulation
BUS Classification

1. PQ buses

P_i known (P_{Di} known, P_{Gi} zero)

Q_i known (Q_{Di} known, Q_{Gi} zero)

$|\overline{V}_i|$ unknown

δ_i unknown

Problem formulation
BUS Classification

2. PV buses

P_i known

Q_i unknown

P_i known (P_{Gi} specified, P_{Di} known)

$|\bar{V}_i|$ known (specified)

Q_i unknown (Q_{Gi} unknown, Q_{Di} known)

δ_i unknown

Problem formulation
BUS Classification

3. Slack bus, generator with large capacity.

$|\overline{V}_1|$ known (specified)

δ_1 known (specified, typically $\delta_1 = 0 \Rightarrow$ reference)

P_1 unknown (P_{D1} known, P_{G1} unknown)

Q_1 unknown (Q_{D1} known, Q_{G1} unknown)

Problem formulation
Variable types and limits

- Power balance

$$\sum_{i=1}^{n} P_{Gi} = \sum_{i=1}^{n} P_{Di} + P_{LOSS}$$

$$\sum_{i=1}^{n} Q_{Gi} = \sum_{i=1}^{n} Q_{Di} + Q_{LOSS}$$

Problem formulation
Variable types and limits

- Variable types

  ✓ Control variables

  P_{Gi} (excepting slack bus)

  Q_{Gi} or $|\overline{V}_i|$

  ✓ Non-control variables

  P_{Di} Q_{Di}

  ✓ State variables

  $|\overline{V}_i|$ δ_i

Problem formulation
Variable types and limits

- Variable limits

  - ✓ Voltage magnitude $\left|\overline{V}_i\right|_{min} \leq \left|\overline{V}_i\right| \leq \left|\overline{V}_i\right|_{max}$

  - ✓ Power angle $\left|\delta_i - \delta_k\right| \leq \left|\delta_i - \delta_k\right|_{max}$ (every existing line)

  - ✓ Power limits

$$P_{Gi,min} \leq P_{Gi} \leq P_{Gi,max}$$

$$Q_{Gi,min} \leq Q_{Gi} \leq Q_{Gi,max}$$

3.The Gauss-Seidel solution technique

Gauss-Seidel solution technique

No lineal system:

$$f(x) = 0 \quad \Rightarrow \quad x = F(x)$$

✓Iteration

$$x^{(r+1)} = F(x^{(r)})$$

✓Stoping rule

$$\left| x^{(r+1)} - x^{(r)} \right| < \varepsilon$$

Gauss-Seidel solution technique

- ✓ Example

$$f(x) = x^2 - 5x + 4 = 0$$

$$x = 0.2x^2 + 0.8$$

$$x^{(r+1)} = 0.2(x^{(r)})^2 + 0.8; \; x^{(0)} = 2$$

$$r = 0, \quad x^{(1)} = 0.2 \times 2^2 + 0.8 = 1.6$$

$$r = 1, \quad x^{(2)} = 0.2 \times 1.6^2 + 0.8 = 1.312$$

$$r = 2, \quad x^{(3)} = 0.2 \times 1.312^2 + 0.8 = 1.1442$$

$$\ldots$$

$$r = 5, \quad x^{(6)} = 1.0103$$

$$r = 6, \quad x^{(7)} = 1.0042$$

$$\ldots$$

$$r = 10, \quad x^{(11)} = 1.0001$$

$$r = 11, \quad x^{(12)} = 1.0000$$

$$r = 12, \quad x^{(13)} = 1.0000$$

Many iterations!

Gauss-Seidel solution technique Algorithm beginning

1) Known

$$P_i \quad i = 2,\ldots,n \quad (\text{PV \& PQ Buses})$$

$$Q_i \quad i = m+1,\ldots,n \quad (\text{PQ Buses})$$

$$\left|\overline{V}_i\right| \quad i = 2,\ldots,m \quad (\text{PV Buses})$$

$$\overline{V}_1 \quad \text{slack bus}$$

$$\left|\overline{V}_i\right|_{min}, \left|\overline{V}_i\right|_{max} \quad i = m+1,\ldots,n \quad (\text{PQ Buses})$$

$$Q_{Gi,min}, Q_{Gi,max} \quad i = 2,\ldots,m \quad (\text{PV Buses})$$

Gauss-Seidel solution technique Algorithm beginning

2) Build $\overline{Y}_{BUS}$

3) Initialize voltages

$$V_i = V_i^0 \qquad i = m+1,\ldots,n$$

$$\delta_i = \delta_i^0 \qquad i = 2,\ldots,n$$

Gauss-Seidel solution technique PQ buses

4) PQ buses

$$\overline{S}_i^* = P_i - jQ_i = \overline{V}_i^* \, \overline{I}_i = \overline{V}_i^* \, \overline{Y}_{ii} \, \overline{V}_i + \overline{V}_i^* \sum_{\substack{k=1 \\ k \neq i}}^{n} \overline{Y}_{ik} \, \overline{V}_k$$

$$\overline{V}_i = \frac{1}{\overline{Y}_{ii}} \left[\frac{P_i - jQ_i}{\overline{V}_i^*} - \sum_{\substack{k=1 \\ k \neq i}}^{n} \overline{Y}_{ik} \, \overline{V}_k \right]$$

$$\overline{A}_i = \frac{P_i - jQ_i}{\overline{Y}_{ii}} \qquad i = m+1, \ldots, n$$

$$\overline{B}_{ik} = \frac{\overline{Y}_{ik}}{\overline{Y}_{ii}} \qquad i = m+1, \ldots, n; \; k = 1, \ldots, n; \; k \neq i$$

Gauss-Seidel solution technique PQ buses

At iteration (r+1) and bus i, the available values of voltages at previous buses are used:

$$\overline{V}_i^{(r+1)} = \frac{\overline{A}_i}{(\overline{V}_i^{(r)})^*} - \sum_{k=1}^{i-1} \overline{B}_{ik} \overline{V}_k^{(r+1)} - \sum_{k=i+1}^{n} \overline{B}_{ik} \overline{V}_k^{(r)}$$

Gauss-Seidel solution technique PV buses

5) PV buses

$$\overline{S}_i^* = P_i - jQ_i = \overline{V}_i^* \overline{I}_i = \overline{V}_i^* \sum_{k=1}^{n} \overline{Y}_{ik} \overline{V}_k$$

$$Q_i = -\Im\left\{ \overline{V}_i^* \sum_{k=1}^{n} \overline{Y}_{ik} \overline{V}_k \right\}$$

Gauss-Seidel solution technique PV buses

At iteration (r+1):

$$Q_i^{(r+1)} = -\Im\left\{(\overline{V}_i^{(r)})^* \sum_{k=1}^{i-1} \overline{Y}_{ik} \overline{V}_k^{(r+1)} + (\overline{V}_i^{(r)})^* \sum_{k=i}^{n} \overline{Y}_{ik} \overline{V}_k^{(r)}\right\}$$

$$\delta_i = \text{angle}\left[\overline{V}_i\right] \qquad \overline{A}_i^{(r+1)} = \frac{P_i - jQ_i^{(r+1)}}{\overline{Y}_{ii}}$$

$$\delta_i^{(r+1)} = \text{angle}\left\{\frac{\overline{A}_i^{(r+1)}}{(\overline{V}_i^{(r)})^*} - \sum_{k=1}^{i-1} \overline{B}_{ik} \overline{V}_k^{(r+1)} - \sum_{k=i+1}^{n} \overline{B}_{ik} \overline{V}_k^{(r)}\right\}$$

Gauss-Seidel solution technique PV buses

Beware of limits!

$$Q_i^{(r+1)} \leq Q_{i,min} \quad \Rightarrow \quad Q_i^{(r+1)} = Q_{i,min} \quad \& \quad i \quad \text{becomesPQ}$$

$$Q_i^{(r+1)} \geq Q_{i,max} \quad \Rightarrow \quad Q_i^{(r+1)} = Q_{i,max} \quad \& \quad i \quad \text{becomesPQ}$$

(more on this below)

Gauss-Seidel solution technique
PV buses

6) Stop criterion

$$\left|V_i^{(r+1)} - V_i^{(r)}\right| < \varepsilon \,; \quad i = 2,\ldots,n$$

6.1) Slack bus power (after convergence)

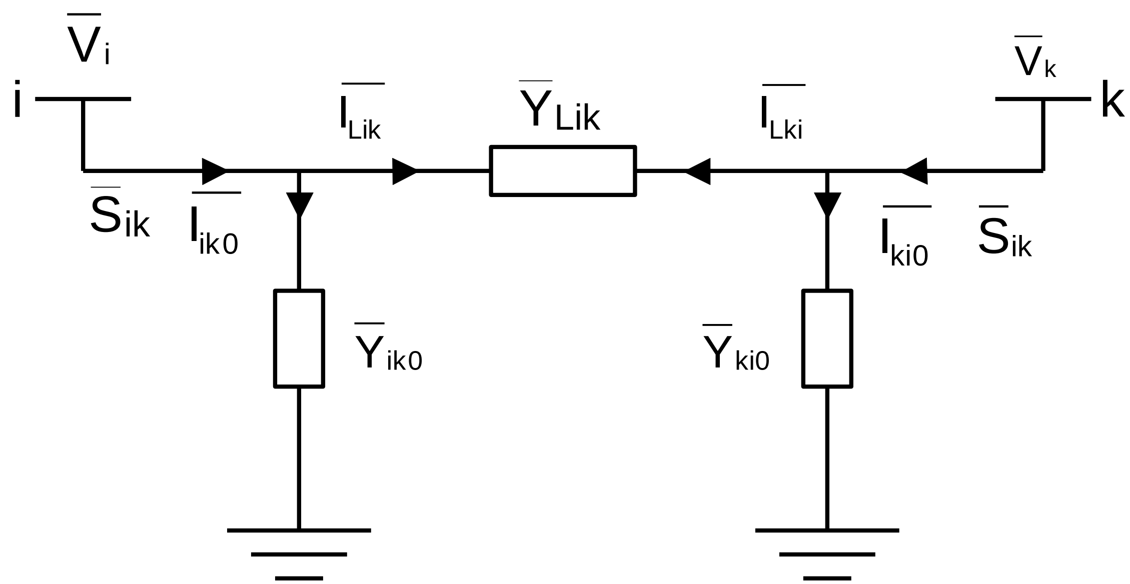
$$P_1 - jQ_1 = \overline{V}_1^{*} \sum_{k=1}^{n} \overline{Y}_{1k} \overline{V}_k$$

Gauss-Seidel solution technique
Algorith final calculations

6.2) Compute line currents (after convergence)

$$\bar{I}_{Lik} = (\bar{V}_i - \bar{V}_k)\,\bar{Y}_{Lik}; \qquad \bar{I}_{ik0} = \bar{V}_i\,\bar{Y}_{ik0}$$

$$\bar{I}_{Lki} = (\bar{V}_k - \bar{V}_i)\,\bar{Y}_{Lik}; \qquad \bar{I}_{ki0} = \bar{V}_k\,\bar{Y}_{ki0}$$

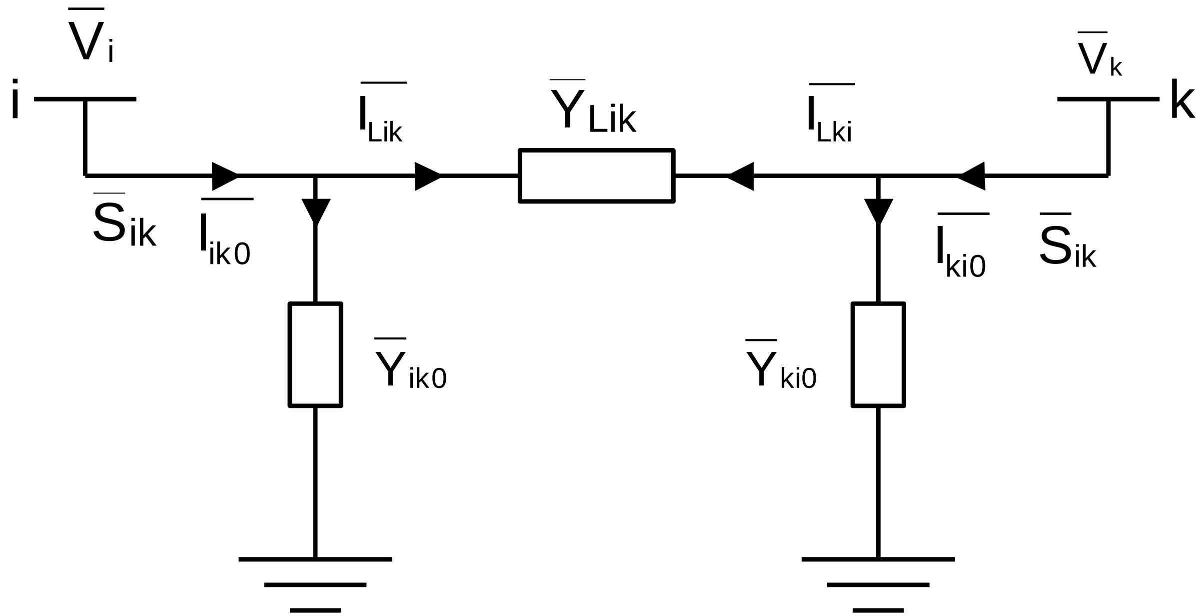


Gauss-Seidel solution technique
Algorith final calculations

6.2) Compute line complex power (after convergence)

$$\overline{S}_{ki} = \overline{V}_k(\overline{V}_k^* - \overline{V}_i^*)\overline{Y}_{Lik}^* + \overline{V}_k \overline{V}_k^* \overline{Y}_{ki0}^*;$$

$$\overline{S}_{ik} = \overline{V}_i(\overline{V}_i^* - \overline{V}_k^*)\overline{Y}_{Lik}^* + \overline{V}_i \overline{V}_i^* \overline{Y}_{ik0}^*;$$



Gauss-Seidel solution technique Algorith final calculations

6.3) Compute losses (after convergence)

$$\bar{S}_{loss,ik} = \bar{S}_{ik} + \bar{S}_{ki} \; ; \quad \forall\, k,i$$

$$\bar{S}_{loss} = \sum_{\forall\, k,i} \bar{S}_{loss,ik}$$

7) If no convergence, go to step 4.

Gauss-Seidel solution technique
Algorithm improvement

- Acceleration factor (in order to decrease the number of iterations):

$$\tilde{\overline{V}}_i^{(r+1)} = \overline{V}_i^{(r)} + \alpha \left[\overline{V}_i^{(r+1)} - \overline{V}_i^{(r)} \right]$$

$\alpha = 1.6$ (generally recommended)

Gauss-Seidel
Matlab code

```
function [Vfinal,angfinal,nite,P,Q,errorplot,tiempo]=Gaussgen(m,n,Ybus,Vmodini,Angini,P,Q,tol,Vmax,Vmin,Qmax,Qmin)
%------------------------------------------------------------------------------------------------------------------------------------------
%-function [Vmod,ang,nite,P,Qerrorplot,tiempo]=Gaussgen(m,n,Ybus,Vmodini,Angini,P,Q,tol,Vmax,Vmin,Qmax,Qmin)
%-Resuelve de forma general problema de carga por el m´etodo de Gauss-Seidel
%-donde:
%-2...m nudos PV; (m=1 cuando no hay nudos PV)
%-n nudos totales
%-Ybus matriz de admitancias
%-Vmodini tensiones iniciales modulo
%-Angini angulos iniciales RADIANES
%-P potencia activa inicial
%-Q potencia reactiva inicial
%-tol tolerancia para error en tension y potencia reactiva
%-Vmax Vmin valores limites aceptables para las tensiones
%-Qmax Qmin valores limites aceptables para las potencias reactivas
%-nite n´umero de iteraciones
%-Vfinal vector con todas las potencias para cada iteraci´on
%-angfinal igual pero con los ´angulos
%-tiempo, tiempo invertido en hacer las operaciones
%------------------------------------------------------------------------------------------------------------------------------------------
%calculo de la matriz B
B=zeros(n,n);
for t=1:n
   for k=1:n
      B(t,k)=Ybus(t,k)/Ybus(t,t);
   end
end
```

Gauss-Seidel
Matlab code 2

```
%calculos los valores en coordenadas cuadrangulares de la tension
V=zeros(n,1);
for a=1:n
    V(a)=Vmodini(a)*exp(i*Angini(a));
end
ang=Angini;
%empieza el bucle:
error=1; %valores iniciales para poder entrar en el bucle
errorQ=1;
nite=0;
tic;
while max(abs(error))>tol | max(abs(errorQ))>tol
    nite=nite+1;
        Vmod=abs(V);
        Vini=V;
        ang=angle(V);
        Qini=Q;
        %calculo las reactivas para los nudos PV
      if m>1
        for l=2:m
            AQ=0;
            for k=1:n
                AQ=AQ+Ybus(l,k)*V(k);
            end
            Q(l)=-imag((V(l)')*AQ);
        end
```

Gauss-Seidel
Matlab code 3

```
%calculo las A para todos los nudos:
    for a=1:n
        A(a)=(P(a)-i*Q(a))/Ybus(a,a);
    end

    %calculo los angulos nudos PV:

     for l=2:m
        Aang=0;
        for k=1:n
            Aang=Aang+B(l,k)*V(k);
        end
        ang(l)=angle(A(l)/((V(2))')-Aang+B(l,l)*V(l));
    end
  end
    for a=1:n
        A(a)=(P(a)-i*Q(a))/Ybus(a,a);
    end
    %ahora actualizo los voltajes otra vez:
    for a=1:n
        V(a)=Vmod(a)*exp(i*ang(a));
    end
    %ahora calculo los nudos PQ
    AV=zeros(n,1);
    for p=m+1:n
        for k=1:n
            AV(p)=AV(p)+B(p,k)*V(k);
        end
        V(p)=A(p)/((V(p))')-AV(p)+B(p,p)*V(p);
    end
    error=Vini-V;
    errorQ=Qini-Q;
    errorplot(1,nite)=norm(abs(error));
    Vfinal(:,nite)=abs(V);
    angfinal(:,nite)=ang*180/pi; %paso a grados
  end
```

Gauss-Seidel

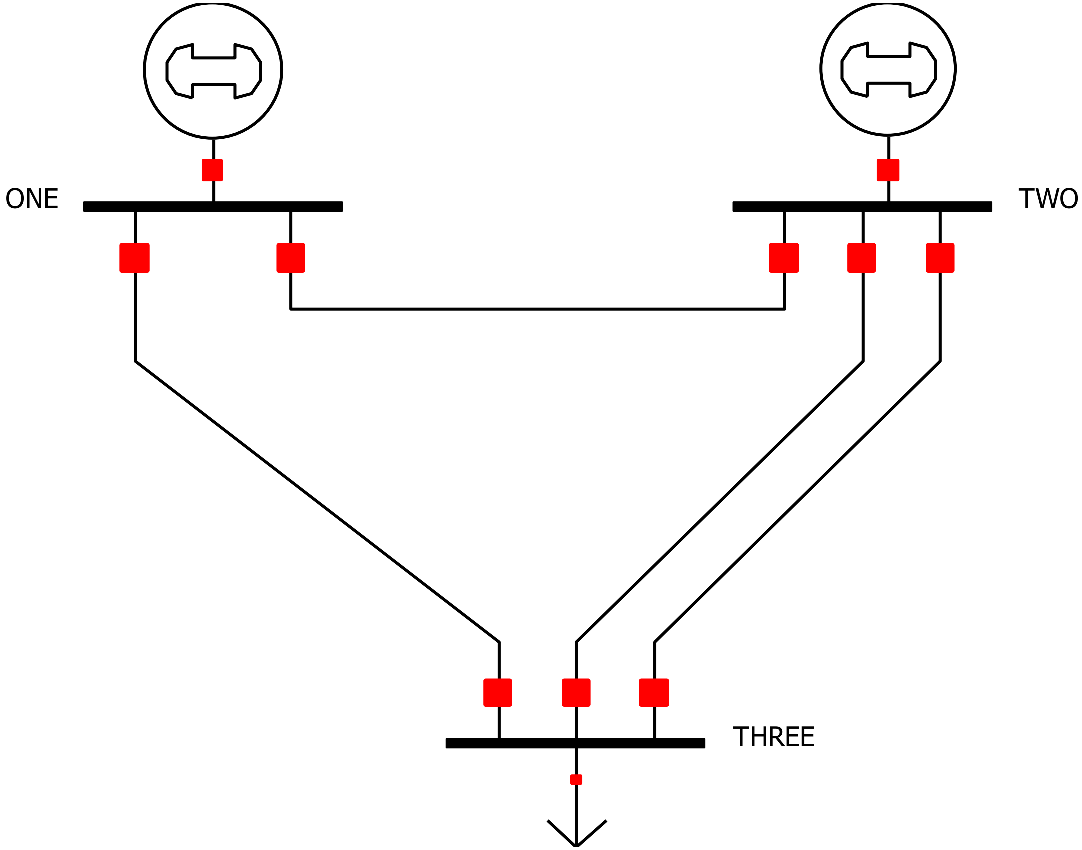
Matlab code 4

```
%calculos los valores de potencia para el nudo slack:
S1=0;
for t=1:n
S1=S1+(V(1)')*(Ybus(1,t)*V(t));
end
P(1)=real(S1);
Q(1)=-imag(S1);
tiempo=toc;
%alerta por si se sobrepasan valores aceptables:
if max(Vmod)>Vmax | min(Vmod)<Vmin
    disp('¡¡SE SOBREPASAN LIMITES TENSIONES!!')
end
if max(Q)>Qmax | min(Q)<Qmin
    disp('¡¡SE SOBREPASAN LIMITES REACTIVA!!')
end
  ang=(180/pi)*angle(V);
   Vmo=abs(V);
   %represento los errores por iteraci´on:

   plot(1:nite,errorplot,'o');grid on;xlabel('iteraci´on');ylabel('error por iteraci´on');title('evoluci´on error tesi´on');

   %-----------------------------------------------------------%
   %                                                           %
   %  Realizado por Carlos Ruiz Mora octubre 2006 %
   %                                                           %
   %-----------------------------------------------------------%
```

Gauss-Seidel example

G-S Example

Solution tolerance is set to 0.1 MVA.



G-S Example

Data below. Base power is S_B=100 MVA:

Bus	Voltage (p.u.)	Power
1	1.02	(slack)
2	1.02	P_G=50 MW
3	-	P_C=100 MW Q_C=60 MVAr

Lin	Impedance (p.u.)
1-2	0.02+0.04j
1-3	0.02+0.06j
2-3	0.02+0.04j (each)

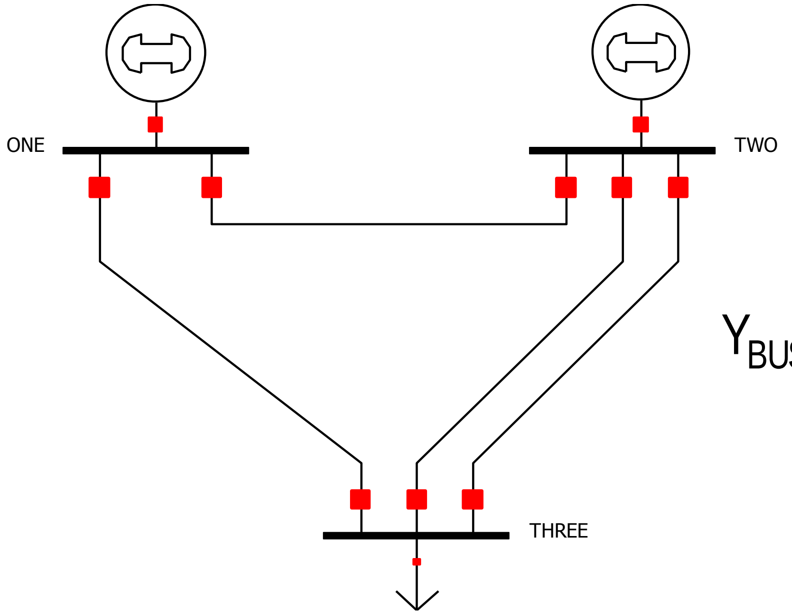
Solution procedure

1. Data and unknown:

Bus	Type	Data	Unknown
1	*Slack*	V_1=1.02 δ_1=0.0	P_1 Q_1
2	*PV*	V_2=1.02 P_2=0.5	δ_2 Q_2
3	*PQ*	P_3=-1.0 Q_2=-0.6	δ_3 V_3

Solution procedure

$$\overline{Y}_{BUS}$$



$$Y_{BUS} = \begin{bmatrix} 15-35j & -10+20j & -5+15j \\ -10+20j & 30-60j & -20+40j \\ -5+15j & -20+40j & 25-55j \end{bmatrix}$$

Solution procedure

3. Voltage magnitude initialization (iteration 0):

$$V_3 = V_3^0 = 1$$ PQ bus

$$\left.\begin{aligned} \delta_2 = \delta_2^0 = 0 \\ \delta_3 = \delta_3^0 = 0 \end{aligned}\right\}$$ All buses but the reference one

Vector form:

$$V^0 = \begin{bmatrix} 1.02 \\ 1.02 \\ 1 \end{bmatrix} \qquad \delta^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Solution procedure

Per iteration:

- PV buses (Q, δ) i=2,...,m (bus 2)

- PQ buses (V, δ) i=m+1,...,n (bus 3)

- Stopping criterios:

a) convergence → S_{slack} and power flows;

b) no converge → new iteration

Solution procedure

4. PV buses: iteration (r+1) :

   →bus 2:

$$Q_2^{(r+1)} = -\Im\left\{(\overline{V}_2^{(r)})^* \overline{Y}_{21}\overline{V}_1^{(r+1)} + (\overline{V}_2^{(r)})^* \overline{Y}_{22}\overline{V}_2^{(r)} + (\overline{V}_2^{(r)})^* \overline{Y}_{23}\overline{V}_3^{(r)}\right\}$$

$$\delta_2 = \text{angle}[\overline{V}_2] \qquad \overline{A}_2^{(r+1)} = \frac{P_2 - jQ_2^{(r+1)}}{\overline{Y}_{22}}$$

Solution procedure

$$\delta_2^{(r+1)} = \text{angle}\left\{ \frac{\overline{A}_2^{(r+1)}}{\left(V_2^{(r+1)}\right)^*} - \overline{B}_{21}\overline{V}_1^{(r+1)} - \overline{B}_{23}\overline{V}_3^{(r)} \right\}$$

where

$$\overline{B}_{ik} = \frac{\overline{Y}_{ik}}{\overline{Y}_{ii}} \quad \text{is a constant}$$

Solution procedure

5. Buses PQ iteration (r+1):

→bus 3:

$$\overline{V}_3^{(r+1)} = \frac{\overline{A}_3}{(\overline{V}_3^{(r)})^*} - \overline{B}_{31}\overline{V}_1^{(r+1)} - \overline{B}_{32}\overline{V}_2^{(r+1)}$$

where

$$\left.\begin{aligned} \overline{A}_3 &= \frac{P_3 - jQ_3}{\overline{Y}_{33}} \\ \overline{B}_{ik} &= \frac{\overline{Y}_{ik}}{\overline{Y}_{ii}} \end{aligned}\right\}$$

Are constants for PQ buses

Solution procedure

6. Stopping criterion:

$$\varepsilon = 10^{-3}$$

$$\left.\begin{array}{c} |V_i^{(r+1)} - V_i^{(r)}| \\ \& \\ |Q_j^{(r+1)} - Q_j^{(r)}| \end{array}\right\} < \varepsilon \qquad i = 2, 3;\ j = 2$$

Solution procedure

If convergence:

6.1) Slack power

$$\overline{S}^*_{slack} = P_1 - jQ_1 = \overline{V}_1^*(\overline{Y}_{11}\overline{V}_1 + \overline{Y}_{12}\overline{V}_2 + \overline{Y}_{13}\overline{V}_3)$$

6.2) Power flows → $\overline{S}_{12}, \overline{S}_{21}, \overline{S}_{13}, \overline{S}_{23}, \overline{S}_{31}, \overline{S}_{32}$

$$\overline{S}_{ik} = \overline{V}_i \ (\overline{V}_i^* - \overline{V}^*{}_k)\overline{Y}^*_{Lik}$$

Solution procedure

6.3) Line losses:

$$\overline{S}_{losss,ik} = \overline{S}_{ik} + \overline{S}_{ki} \qquad k,i = 1,2,3$$

$$\overline{S}_{loss} = \overline{S}_{loss,12} + \overline{S}_{loss,13} + \overline{S}_{loss,23}$$

7. If no coveergence, the procedure continues in Step 4.

Implementation

MATLAB:

Solution

11 iterations needed to attain the solution

Iteration **(pu)**	1	2	3	...	10	11
P_1, Q_1(slack)	-	-	-	...	-	***0.5083*** ***0.0716***
P_2	0.5	0.5	0.5	...	0.5	0.5
Q_2	***0.81***	***0.4084***	***0.4696***	***...***	***0.5493***	***0.5501***
P_3	-1.0	-1.0	-1.0	...	-1.0	-1.0
Q_3	-0.6	-0.6	-0.6	...	-0.6	-0.6

Solution

Iteration	1	2	3	...	10	11
V_1	1.02	1.02	1.02	...	1.02	1.02
$\delta_1(^o)$	0	0	0	...	0	0
V_2	1.02	1.02	1.02	...	1.02	1.02
$\delta_2(^o)$	***0.0675***	***-0.1596***	***-0.2885***	***...***	***-0.4667***	***-0.4685***
V_3	***1.0041***	***1.0042***	***1.0043***	***...***	***1.0043***	***1.0043***
$\delta_3(^o)$	***-0.5746***	***-0.7336***	***-0.8278***	***...***	***-0.9580***	***-0.9593***

Solution

Errors for |V| & |Q|:

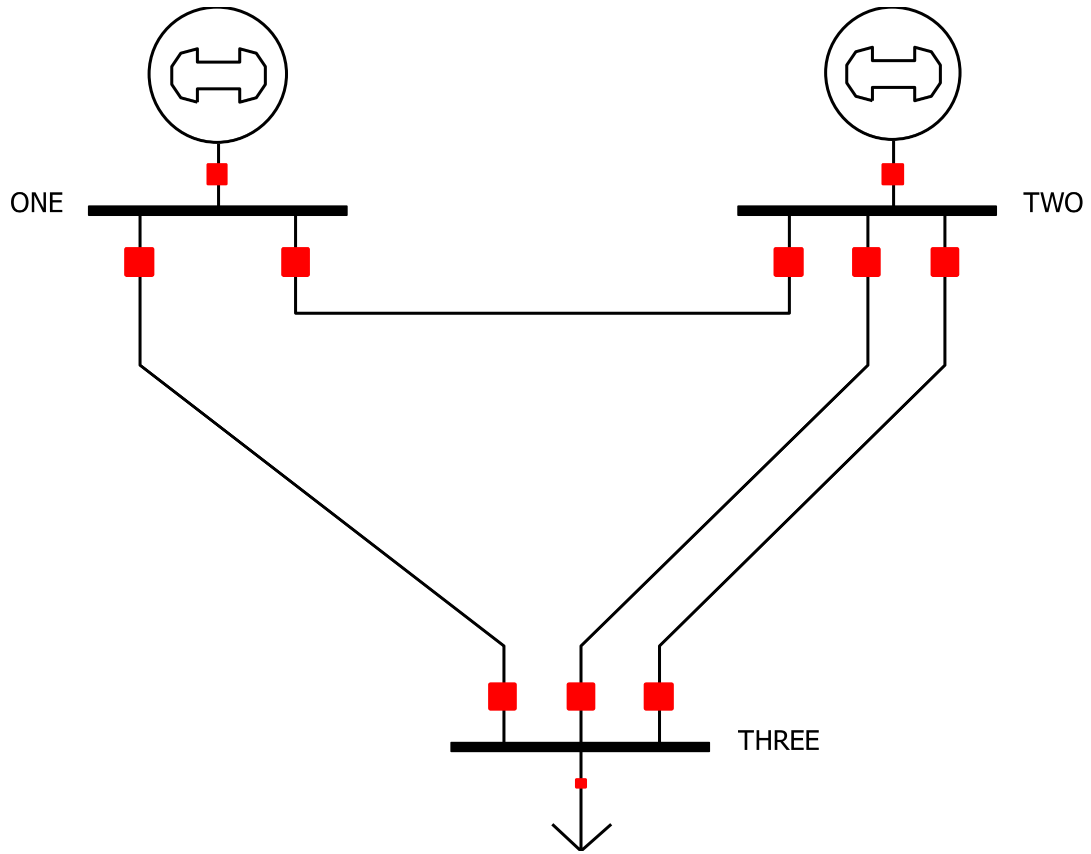
Iteration	1	2	3	...	10	11
Max. Error V	0.0041	$9.68 \cdot 10^{-5}$	$5.05 \cdot 10^{-5}$	...	$1.15 \cdot 10^{-6}$	$6.74 \cdot 10^{-7}$
Max. Error Q	0.8160	0.4076	0.0612	...	0.0014	$8.11 \cdot 10^{-4}$

Final Solution

Bus	P (MW)	Q (MVAr)	$\overline{V}$ (pu)
1	50.83	7.16	$1.02 \angle 0^o$
2	50.00	55.01	$1.02 \angle -0.4685^o$
3	-100	-60.00	$1.0043 \angle -0.9593^o$

Checking

Checking using Power-World:



Checking

	Bus 1		**Bus 2**		**Bus 3**	
Solution	**G-S**	**PW**	**G-S**	**PW**	**G-S**	**PW**
V ***(pu)***	1.02	1.02	1.02	1.02	1.0043	1.0043
δ(º)	0	0	-0.4685	-0.4710	-0.9593	-0.9612
P ***(****MW****)***	50.83	50.98	50.00	50.00	-100	-100
Q ***(****MVAr****)***	7.16	7.10	55.01	55.12	-60	-60

Checking

Variable	*V*	*δ*	*P*	*Q*
Error Max. (%)	0 %	0.53 %	0.29%	0.84 %

4. The Newton-Raphson solution technique

The Newton-Raphson solution technique
Introduction

- One variable

$$f(x) = 0, \qquad f(x^{(0)}) \neq 0, \quad f(x^{(0)} + \Delta x^{(0)}) = 0$$

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

$$\Delta x^{(0)} = -\frac{f(x^{(0)})}{\left(\frac{df}{dx}\right)^{(0)}}; \quad x^{(1)} = x^{(0)} + \Delta x^{(0)}$$

$$x^{(r+1)} = x^{(r)} - \frac{f(x^{(r)})}{\left(\frac{df}{dx}\right)^{(r)}}$$

The Newton-Raphson solution technique
Introduction

- Example

$$f(x) = x^2 - 5x + 4 = 0; \frac{df(x)}{dx} = 2x - 5$$

$$x^{(r+1)} = x^{(r)} - \frac{(x^{(r)})^2 - 5x^{(r)} + 4}{2x^{(r)} - 5}; x^{(0)} = 0.5$$

$$x^{(1)} = 0.5 - \frac{0.5^2 - 5 \times 0.5 + 4}{2 \times 0.5 - 5} = 0.9375$$

$$x^{(2)} = 0.9375 - \frac{0.9375^2 - 5 \times 0.9375 + 4}{2 \times 0.9375 - 5} = 0.9988$$

$$x^{(3)} = 0.9988 - \frac{0.9988^2 - 5 \times 0.9988 + 4}{2 \times 0.9988 - 5} = 1.0000$$

Fast!

The Newton-Raphson solution technique
General formulation

- General case

$$f(x) = 0 : R^n \rightarrow R^n \qquad f(x^{(0)}) \neq 0, \quad f(x^{(0)} + \Delta x^{(0)}) = 0$$

$$f(x^{(0)} + \Delta x) = f(x^{(0)}) + J^{(0)} \times \Delta x^{(0)} = 0$$

$$\Delta x^{(0)} = -\left[J^{(0)}\right]^{-1} f(x^{(0)}) \longleftarrow$$

$$x^{(1)} = x^{(0)} + \Delta x^{(0)}$$

$$x^{(r+1)} = x^{(r)} - \left[J^{(r)}\right]^{-1} f(x^{(r)})$$

The Newton-Raphson solution technique
General formulation

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \qquad f = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_n \end{bmatrix} \qquad J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

The Newton-Raphson solution technique
Load flow case

$$\left[x^{(r)}\right]=\begin{bmatrix}\delta_2^{(r)}\\ \delta_3^{(r)}\\ \vdots\\ \delta_n^{(r)}\\ V_{m+1}^{(r)}\\ \vdots\\ V_{n-1}^{(r)}\\ V_n^{(r)}\end{bmatrix}$$

$$f_{iP}(\bullet)\equiv V_i\times\sum_{k=1}^{n}V_k(G_{ik}\cos\delta_{ik}+B_{ik}\sin\delta_{ik})$$

$$f_{iQ}(\bullet)\equiv V_i\times\sum_{k=1}^{n}V_k(G_{ik}\sin\delta_{ik}-B_{ik}\cos\delta_{ik})$$

$$P_i(\cdot)\equiv f_{iP},\quad P_i=P_{i,sp}$$

$$Q_i(\cdot)\equiv f_{iQ},\quad Q_i=Q_{i,sp}$$

$$\delta_2,\ldots,\delta_n;V_{m+1},\ldots,V_n$$

The Newton-Raphson solution technique Load flow case

Using Taylor:

$$P_{i,sp} \approx f_{iP}^{(r)} + \left[\frac{\partial f_{iP}}{\partial \delta_2}\right]^{(r)} \Delta\delta_2^{(r)} + \ldots + \left[\frac{\partial f_{iP}}{\partial \delta_n}\right]^{(r)} \Delta\delta_n^{(r)} + \left[\frac{\partial f_{iP}}{\partial V_{m+1}}\right]^{(r)} \Delta V_{m+1}^{(r)} + \ldots + \left[\frac{\partial f_{iP}}{\partial V_n}\right]^{(r)} \Delta V_n^{(r)}$$

$$Q_{i,sp} \approx f_{iQ}^{(r)} + \left[\frac{\partial f_{iQ}}{\partial \delta_2}\right]^{(r)} \Delta\delta_2^{(r)} + \ldots + \left[\frac{\partial f_{iQ}}{\partial \delta_n}\right]^{(r)} \Delta\delta_n^{(r)} + \left[\frac{\partial f_{iQ}}{\partial V_{m+1}}\right]^{(r)} \Delta V_{m+1}^{(r)} + \ldots + \left[\frac{\partial f_{iQ}}{\partial V_n}\right]^{(r)} \Delta V_n^{(r)}$$

The increments below should be 0:

$$\Delta P_i^{(r)} = P_{i,sp} - f_{iP}^{(r)}; \quad \Delta Q_i^{(r)} = Q_{i,sp} - f_{iQ}^{(r)}$$

The Newton-Raphson solution technique
Load flow case

Matrix notation:

$$
\begin{bmatrix} \Delta P_2^{(r)} \\ \cdots \\ \Delta P_2^{(r)} \\ \Delta Q_{m+1}^{(r)} \\ \cdots \\ \Delta Q_n^{(r)} \end{bmatrix}
=
\begin{bmatrix}
\left[\dfrac{\partial f_{2P}}{\partial \delta_2}\right]^{(r)} & \cdots & \cdots & \left[\dfrac{\partial f_{2P}}{\partial V_n}\right]^{(r)} \\
\cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
\left[\dfrac{\partial f_{nQ}}{\partial \delta_2}\right]^{(r)} & \cdots & \cdots & \left[\dfrac{\partial f_{nQ}}{\partial \delta_n}\right]^{(r)}
\end{bmatrix}
\begin{bmatrix} \Delta \delta_2^{(r)} \\ \cdots \\ \Delta \delta_n^{(r)} \\ \Delta V_{m+1}^{(r)} \\ \cdots \\ \Delta V_n^{(r)} \end{bmatrix}
$$

The Newton-Raphson solution technique
Jacobian matrix

- PQ buses generate 2 Jacobian rows corresponding to ΔP and ΔQ
- PV buses generate 1 Jacobian row corresponding to ΔP

The Newton-Raphson solution technique
Jacobian elements

- Jacobian dimension

$$2 \times N_{PQ} + N_{PV}$$

N_{PQ} Number of PQ buses

N_{PV} Number of PV buses

The Newton-Raphson solution technique Jacobian elements

- Jacobian dimension

$$\begin{bmatrix} \Delta P^{(r)} \\ \Delta Q^{(r)} \end{bmatrix} = \begin{bmatrix} H^{(r)} & N^{(r)} \\ J^{(r)} & L^{(r)} \end{bmatrix} \times \begin{bmatrix} \Delta\delta^{(r+1)} \\ \Delta V^{(r+1)} \big/ V^{(r)} \end{bmatrix}$$

$\dfrac{\Delta V^{(r+1)}}{V^{(r)}}$ instead of $\Delta V^{(r+1)}$

for improving computational efficiency

The Newton-Raphson solution technique Jacobian elements

- $\forall m \neq k$

$$H_{km} = \frac{\partial P_k}{\partial \delta_m} = V_k V_m (G_{km} \sin\delta_{km} - B_{km} \cos\delta_{km})$$

$$N_{km} = V_m \frac{\partial P_k}{\partial V_m} = V_k V_m (G_{km} \cos\delta_{km} + B_{km} \sin\delta_{km})$$

$$J_{km} = \frac{\partial Q_k}{\partial \delta_m} = -V_k V_m (G_{km} \cos\delta_{km} + B_{km} \sin\delta_{km})$$

$$L_{km} = V_m \frac{\partial Q_k}{\partial V_m} = V_k V_m (G_{km} \sin\delta_{km} - B_{km} \cos\delta_{km})$$

The Newton-Raphson solution technique Jacobian elements

- $k = m$

$$H_{kk} = \frac{\Delta P_k}{\Delta \delta_k} = -Q_k - B_{kk} V_k^2$$

$$N_{kk} = V_k \frac{\partial P_k}{\partial V_k} = P_k + G_{kk} V_k^2$$

$$J_{kk} = \frac{\partial Q_k}{\partial \delta_k} = P_k - G_{kk} V_k^2$$

$$L_{kk} = V_k \frac{\partial Q_k}{\partial V_k} = Q_k - B_{kk} V_k^2$$

- Note :

$$\delta_{km} \equiv \delta_k - \delta_m$$

$$\overline{Y}_{km} \equiv G_{km} + jB_{km}$$

The Newton-Raphson solution technique
Solution outline

1. Build $\bar{Y}_{BUS}$
2. Specify $\delta_1 = 0$

   $P_i \qquad i = 2, \ldots, n$

   $Q_i \qquad i = m+1, \ldots, n$

   $V_i \qquad i = 2, \ldots, m$

   $|\bar{V}_1|$

3. Initialize $\begin{cases} \delta_i, \; i = 2, \ldots, n \\ |\bar{V}_i|, \; i = m+1, \ldots, n \end{cases}$

The Newton-Raphson solution technique
Solution outline

4. Compute $\Delta P_i^{(r)}$ (for PV & PQ Buses), $\Delta Q_i^{(r)}$ (for PQ Buses)

5. If $\Delta P_i^{(r)} \leq \varepsilon_P$ & $\Delta Q_i^{(r)} \leq \varepsilon_Q$

$$\text{then compute} \quad \begin{cases} 1) & P_1 + jQ_1 \\ 2) & \text{line flows} \\ 3) & \text{Stop} \end{cases}$$

else go on

6. Compute submatrices $H^{(r)}, N^{(r)}, J^{(r)}, L^{(r)}$

The Newton-Raphson solution technique
Solution outline

7. Solve

$$\begin{bmatrix} \Delta\delta^{(r+1)} \\ \Delta V^{(r+1)} / V^{(r)} \end{bmatrix} = \begin{bmatrix} H^{(r)} & N^{(r)} \\ J^{(r)} & L^{(r)} \end{bmatrix}^{-1} \begin{bmatrix} \Delta P^{(r)} \\ \Delta Q^{(r)} \end{bmatrix}$$

8. Update

$$\delta^{(r+1)} = \delta^{(r)} + \Delta\delta^{(r+1)} \quad \text{PV \& PQ Buses}$$

$$V^{(r+1)} = V^{(r)} + \Delta V^{(r+1)} \quad \text{PV Buses}$$

9. Go to step 4

The Newton-Raphson, Matlab Code

```
function [V_modulo, V_fase, P, Q, N_iter, T_calculo, Error_p] = nraphson(V_modulo_ini, V_fase_ini, P_ini, Q_ini, Y, npv)

%
% [V_modulo, V_fase, P, Q, N_iter, T_calculo, Error_p] = nraphson(V_modulo_ini, V_fase_ini, P_ini, Q_ini, Y, npv)
%
% Obtencion de indices de nodos:    1           ->  SLACK;
%                                   2,...,M     ->  PV;
%                                   M+1,...,N   ->  PQ

n = length(V_modulo_ini);
m = npv + 1;

% Parametros del Metodo
Tol     = 0.0001;          % Tolerancia del metodo (Perdida de potencia).
Error_p = 1;               % Error inicial (A un valor mayor que Tol).
N_iter  = 0;               % Numero de iteraciones.

% Valores iniciales
V_modulo= V_modulo_ini';
V_fase  = V_fase_ini';
P  = P_ini';
Q = Q_ini';
j=sqrt(-1);
```

The Newton-Raphson, Matlab Code

```
tic;
while (Error_p > Tol)                        % Bucle principal  (Se permiten 50 iteraciones como mucho).
   if (N_iter >50)
      error('Demasiadas iteraciones');
      break
   end
   V = V_modulo.*exp(j*V_fase);              % Expresion compleja de la tension
   S = V.*conj(Y*V);                         % Expresion compleja de la potencia
   DP = P(2:n)-real(S(2:n));                 % Incremento de potencia activa (nudos PV y PQ)
   DQ = Q(m+1:n)-imag(S(m+1:n));             % Incremento de potencia reactiva (nudos PQ)

   PQ = [DP ; DQ];
   Error_p = norm(PQ,2);                     % Error en esa iteracion

   DS_DA = diag(V)*conj(Y*j*diag(V)) + diag(conj(Y*V))*j*diag(V);
   DS_DV = diag(V)*conj(Y*diag(V./V_modulo)) + diag(conj(Y*V))*diag(V./V_modulo);

% Construccion del Jacobiano
J = [real(DS_DA(2:n   , 2:n)) real(DS_DV(2:n   , m+1:n))
          imag(DS_DA(m+1:n , 2:n)) imag(DS_DV(m+1:n , m+1:n))]

   dx=J\PQ;                                  % indices: 1...n-1 = fases en PV y PQ; n...final = modulos en PQ
```

The Newton-Raphson, Matlab Code

```
V_fase   (2:n)  = V_fase(2:n)     + dx(1:n-1);            % Actualizamos la fase de las tensiones (nudos PV y PQ)
   V_modulo (m+1:n)= V_modulo(m+1:n) + dx(n:end);   % Actualizamos el modulo de las tensiones (nudos PQ)


   N_iter  = N_iter + 1;                                   % Incremento el numero de iteraciones

  disp('Pulse una tecla para continuar')
  pause
end
P=real(S);                                                  % Calculo de la potencia activa
Q=imag(S);                                                  % Calculo de la potencia reactiva
V_fase=V_fase*180/pi;                                       % Paso de Radianes a grados
T_calculo=toc;

%              **** ENTRADAS ****
%
%  V_modulo_ini = Modulo de la tension para comenzar a iterar (conocidos en SLACK y PV).
%  V_fase_ini   = Fase de la tension para comenzar a iterar (conocido en SLACK).
%  P_ini        = Potencia activa en los nodos (conocido en PV y PQ).
%  Q_ini        = Potencia reactica en los nodos (conocido en PQ).
%  Y            = Matriz de admitancias nodales.
```

The Newton-Raphson, Matlab Code

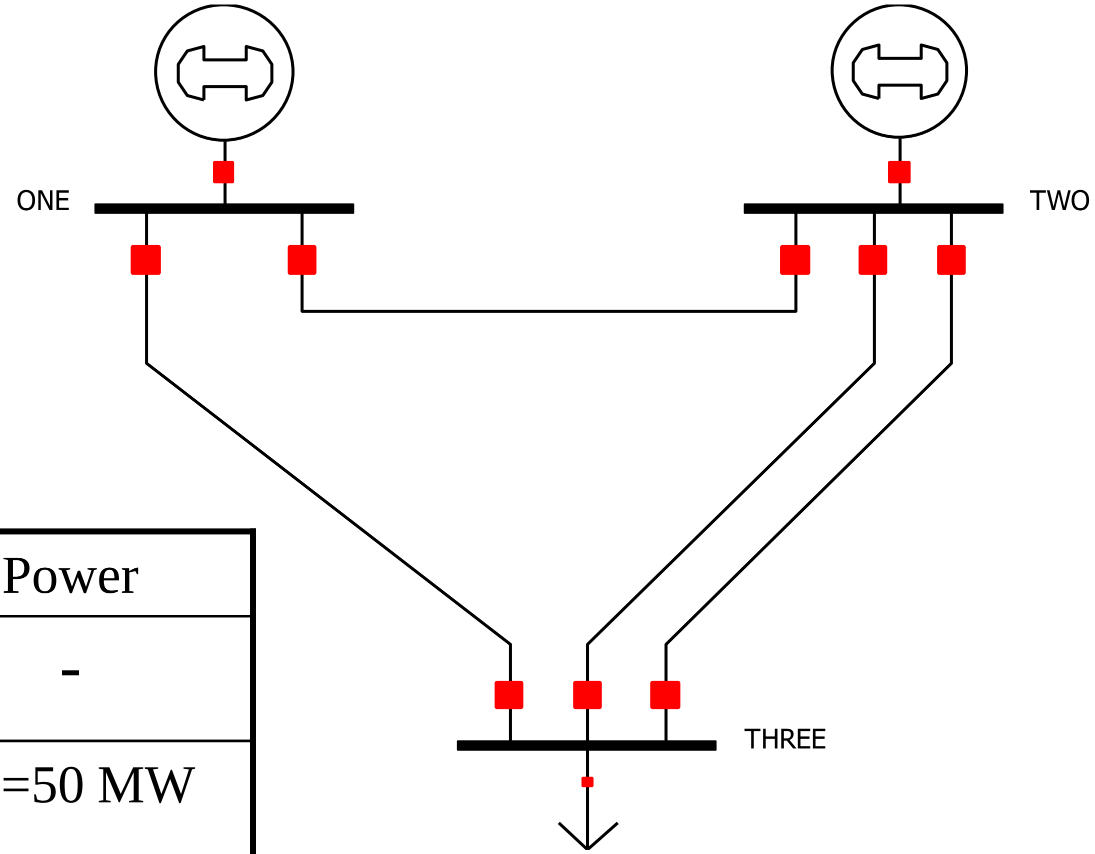
```
%             **** SALIDAS  ****
%
%  V_modulo     = Modulo de la tension en todos los nodos.
%  V_fase       = Fase de la tension en todos los nodos.
%  P            = Potencia activa en todos los nodos.
%  Q            = Potencia reactiva en todos los nodos.
%  N_iter       = Numero de iteraciones.
%  T_calculo    = Tiempo de calculo.
%  Error_p      = Error.
%
%
%             **** OBSERVACIONES ****
%
%  Los nodos deben estar ordenador asi:
%
%    * 1       : SLACK.
%    * 2...m   : PV.
%    * m+1...n : PQ.
%
%
%             **** FECHA Y AUTOR ****
%
%  Laura Laguna - Octubre de 2005
```

Ejemplo resuelto por el método de Newton-Raphson

Newton-Raphson Example

- Checking with Matlab and PowerWorld

Newton-Raphson Example



Bus	Voltage p.u	Power
1	1.02	-
2	1.02	P_G=50 MW
3	-	P_C= 100 MW Q_C=60 MVAr

Newton-Raphson Example

Data:

Line	Impedance p.u.
1-2	0.02+0.04j
1-3	0.02+0.06j
2-3	0.02+0.04j each

Newton-Raphson Example

- 3 buses:
  - Bus1: Slack
  - Bus 2: PV
  - Bus 3: PQ

- Voltage magnitude at bus 3 initialized at 1.02.

- Angles intitialized to zero.

Newton-Raphson Example

Y bus:

$$\overline{Y} = \begin{pmatrix} 15.0000 - 35.0000j & -10.0000 + 20.0000j & -5.0000 + 15.0000j \\ -10.0000 + 20.0000j & 30.0000 - 60.0000j & -20.0000 + 40.0000j \\ -5.0000 + 15.0000j & -20.0000 + 40.0000j & 25.0000 - 55.0000j \end{pmatrix}$$

Intitialization:

$$\delta_2 = \delta_3 = 0$$

$$V_3 = 1.02$$

Newton-Raphson Example

Residuals:

$$\Delta P_i = P_i^{esp} - V_i \sum_{j=1}^{3} V_j (G_{ij} \cos\delta_{ij} + B_{ij} sen(\delta_{ij})) \quad i = 2,3$$

$$\Delta Q_i = Q_i^{esp} - V_i \sum_{j=1}^{3} V_j (G_{ij} sen\delta_{ij} - B_{ij} \cos(\delta_{ij})) \quad i = 3$$

Newton-Raphson Example

Checking:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.3624 \cdot 10^{-14} \\ 0 \\ 7.2475 \cdot 10^{-15} \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} 0.5 \\ -1 \\ -0.6 \end{bmatrix}$$

No tolerance satisfied: the process continues.

Newton-Raphson Example

Jacobian:

$$J = \begin{pmatrix} H_{22} & H_{23} & N_{23} \\ H_{32} & H_{33} & N_{33} \\ M_{32} & M_{33} & L_{33} \end{pmatrix} \longrightarrow J = \begin{pmatrix} 62.4240 & -41.6160 & -20.4000 \\ -41.6160 & 57.2220 & 25.5000 \\ 20.8080 & -26.0100 & 56.1000 \end{pmatrix}$$

Newton-Raphson Example

First iteration:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.4557 \\ -0.9406 \\ -0.0154 \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0046 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4557 \\ -0.9406 \end{pmatrix}$$

Ejemplo por Newton-Raphson

Residuals:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.4958 \\ -0.9835 \\ -0.5874 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - 0.4958 \\ -1 - (-0.9835) \\ -0.6 - (-0.5874) \end{bmatrix} = \begin{bmatrix} 0.0042 \\ -0.0165 \\ -0.0126 \end{bmatrix}$$

No convergence.

Newton-Raphson Example

Jacobian for iteration 2:

$$J = \begin{pmatrix} 61.8860 & -41.1614 & -20.0540 \\ -40.8145 & 56.0994 & 24.1371 \\ 20.8409 & -26.2162 & 54.6707 \end{pmatrix}$$

Newton-Raphson Example

State variables at iteration 2

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.0154 \\ -0.0206 \\ -3.0024\cdot 10^{-4} \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0043 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4710 \\ -0.9612 \end{pmatrix}$$

Newton-Raphson Example

Residuals:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.5000 \\ -1.0000 \\ -0.6000 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - 0.5000 \\ -1 - (-1.0000) \\ -0.6 - (-0.6000) \end{bmatrix} = \begin{bmatrix} 0.0664 \cdot 10^{-5} \\ -0.5454 \cdot 10^{-5} \\ -4.9698 \cdot 10^{-6} \end{bmatrix}$$

Tolerance OK.

Newton-Raphson Example

Jacobian iteration 3:

$$J = \begin{pmatrix} 61.8860 & -41.1614 & -20.0540 \\ -40.8145 & 56.0994 & 24.1371 \\ 20.8409 & -26.2162 & 54.6707 \end{pmatrix}$$

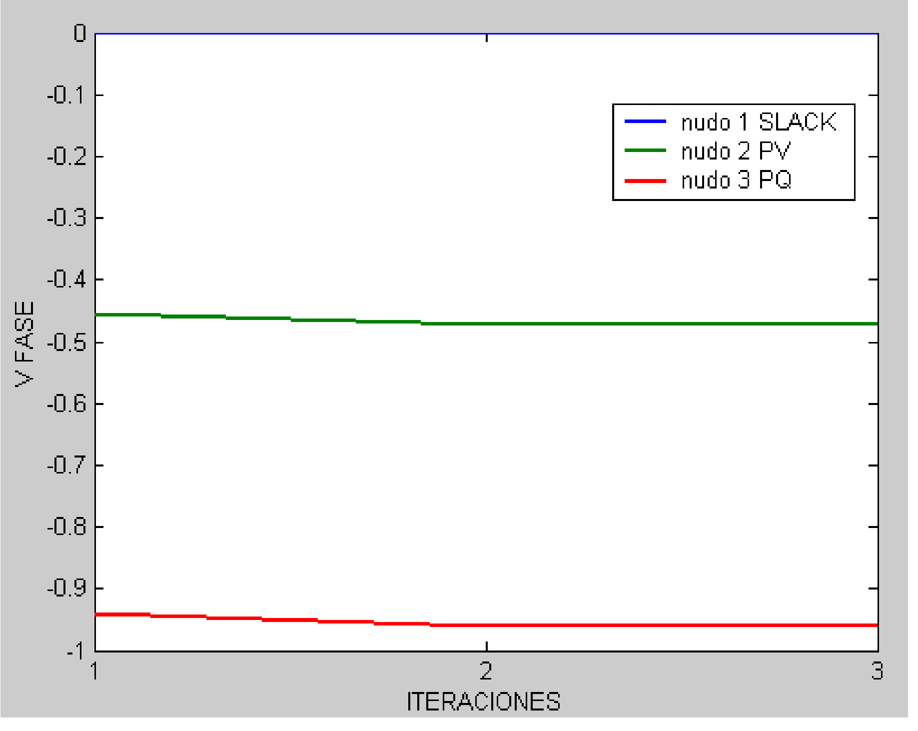
$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.6458 \cdot 10^{-5} \\ -0.7534 \cdot 10^{-5} \\ -1.1106 \cdot 10^{-7} \end{bmatrix} \quad V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0043 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4710 \\ -0.96810 \end{pmatrix}$$

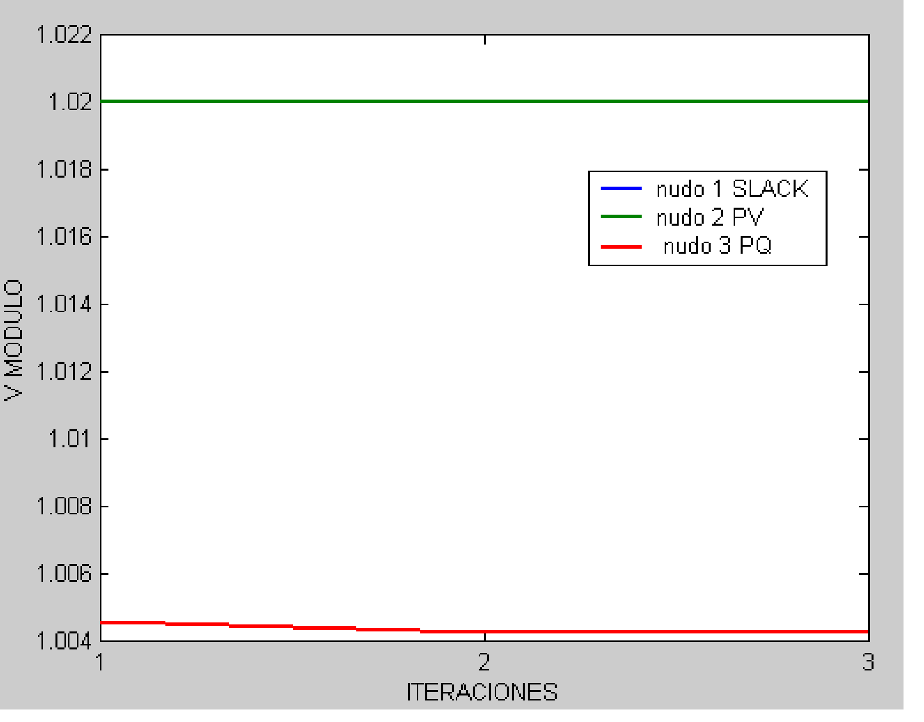
Newton-Raphson Example

Final power values:

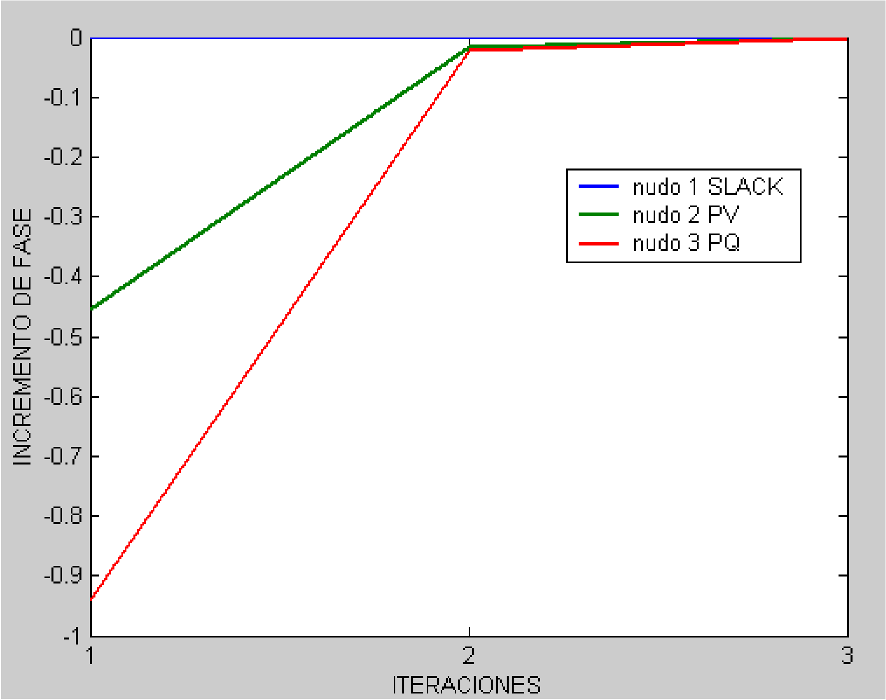
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 0.5098 \\ 0.5000 \\ -1.0000 \end{bmatrix} \qquad \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} 0.0710 \\ 0.5513 \\ -0.6000 \end{bmatrix}$$

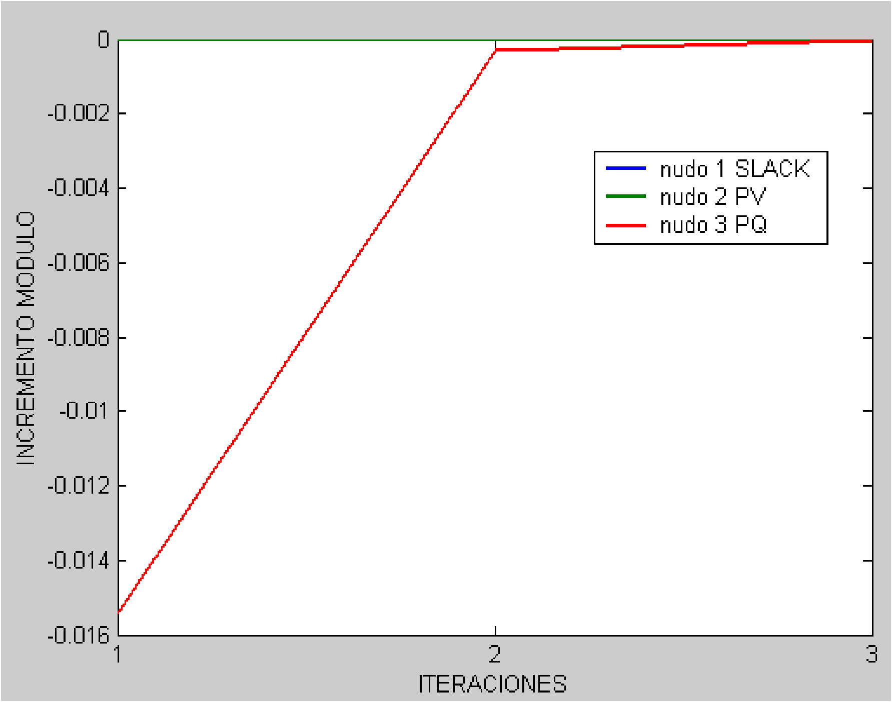
Newton-Raphson Example convergence





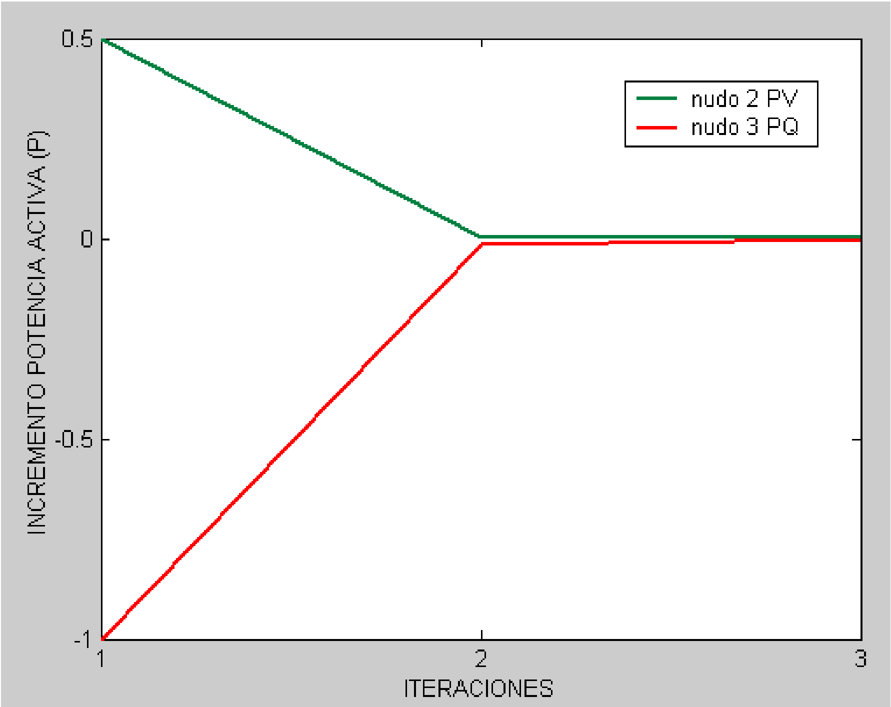
Newton-Raphson Example
convergence

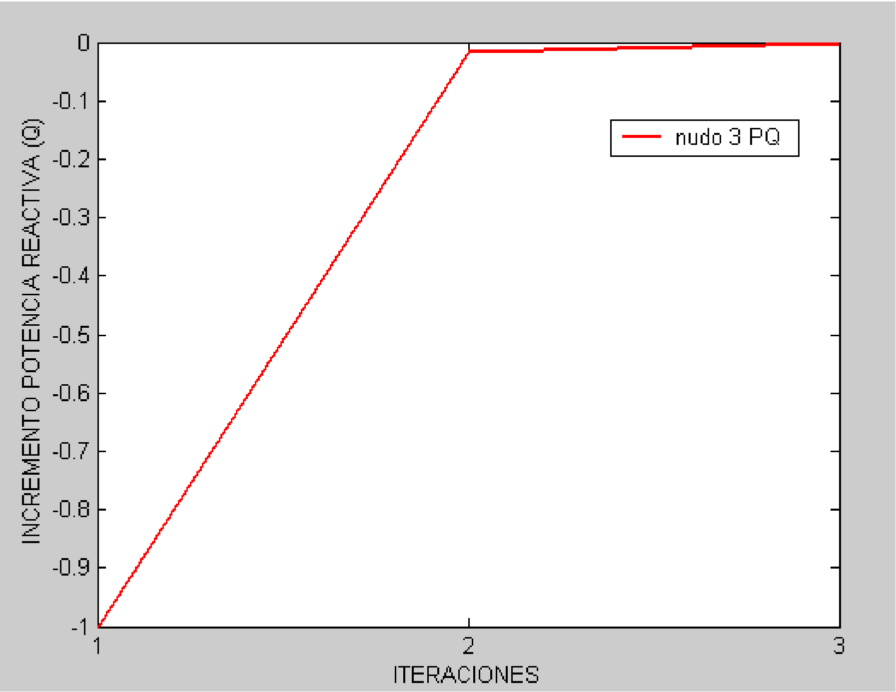




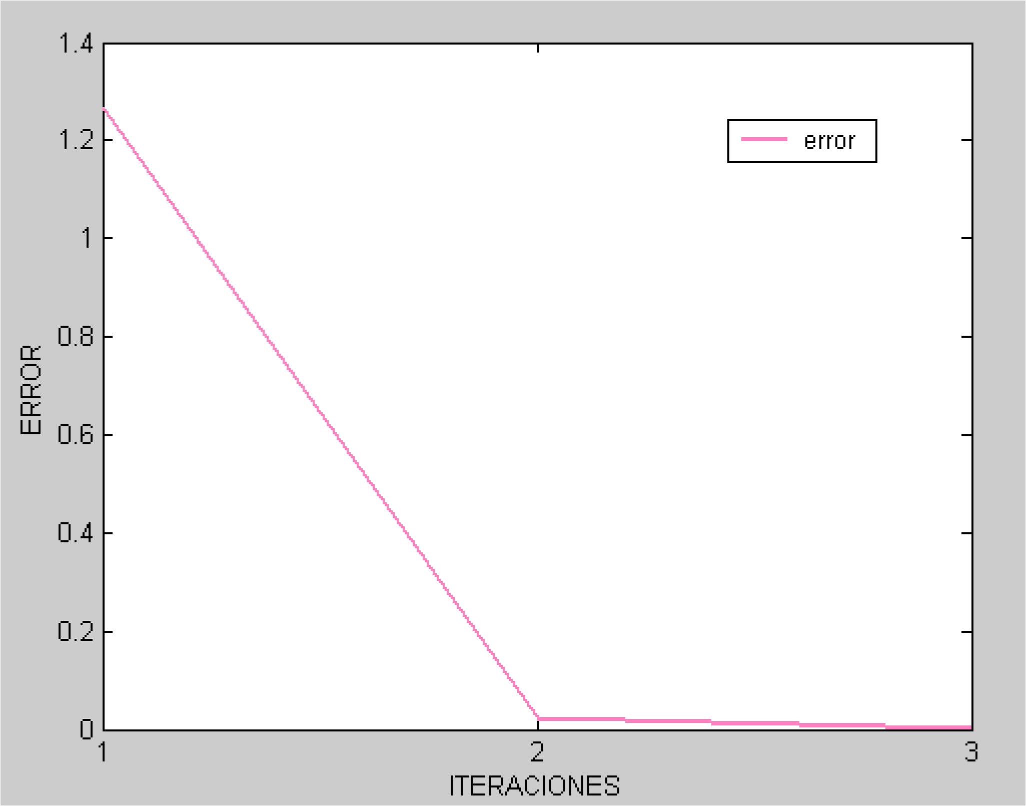
Newton-Raphson Example

convergence

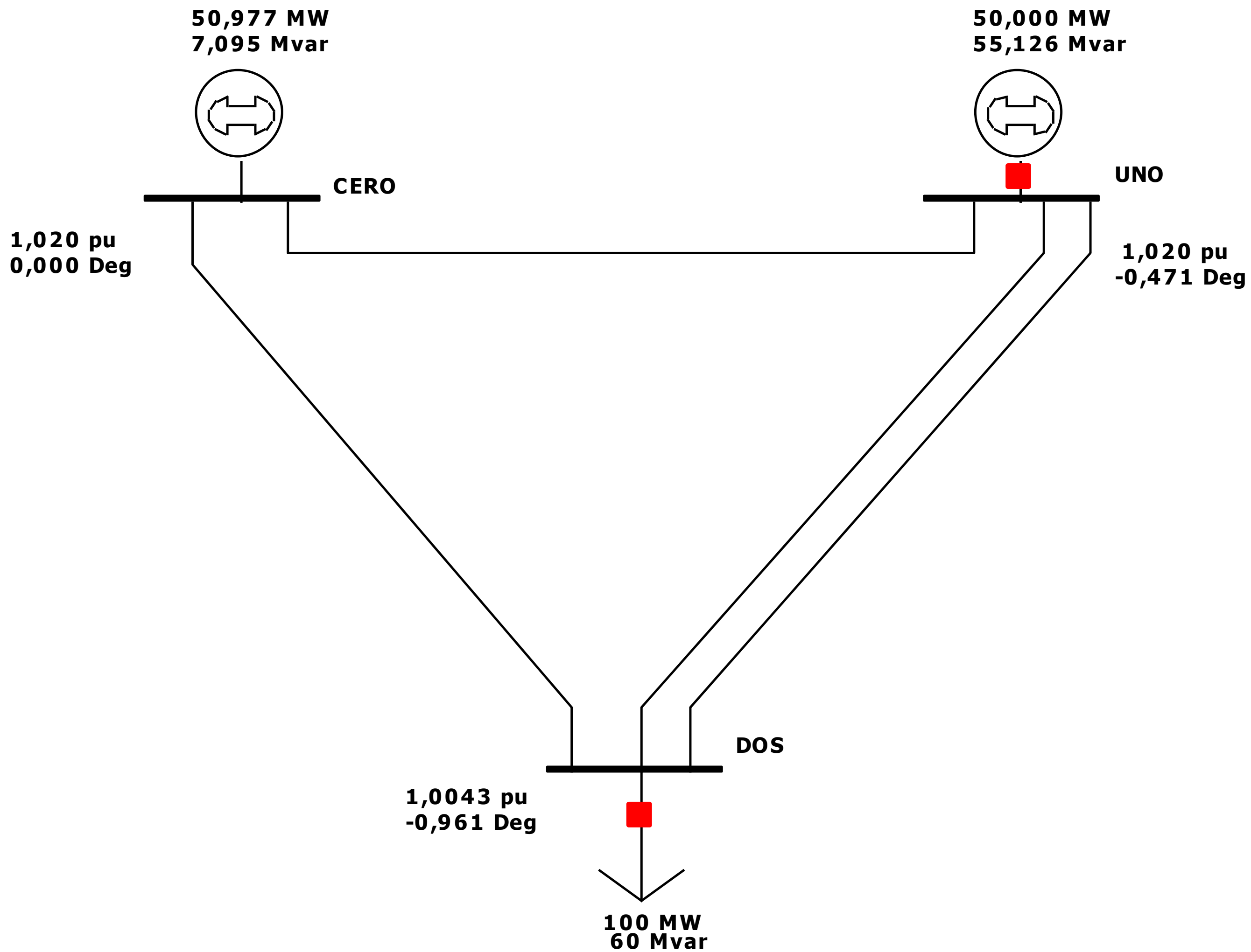




Newton-Raphson Example
convergence



Power World


50,977 MW
7,095 Mvar
50,000 MW
55,126 Mvar
CERO
UNO
1,020 pu
0,000 Deg
1,020 pu
-0,471 Deg
DOS
1,0043 pu
-0,961 Deg
100 MW
60 Mvar

Newton-Raphson Example

Bus	V(p.u)	δ	P	Q
0 (PW)	1.02	0	50.9763	7.0955
0	1.02	0	50.9755	7.0954
1 (PW)	1.02	-0.4710	50	55.1256
1	1.02	-0.4710	50.0006	55.1228
2 (PW)	1.0043	-0.9612	-100	-60
2	1.0043	-0.9612	-99.9989	-59.9970

Newton-Raphson Example

- Largest error below 0.1 MVA.
- More effective technique than Gauss Seidel.
- Convergence is fast (if adequate initialization).

Including tap-changing transformers

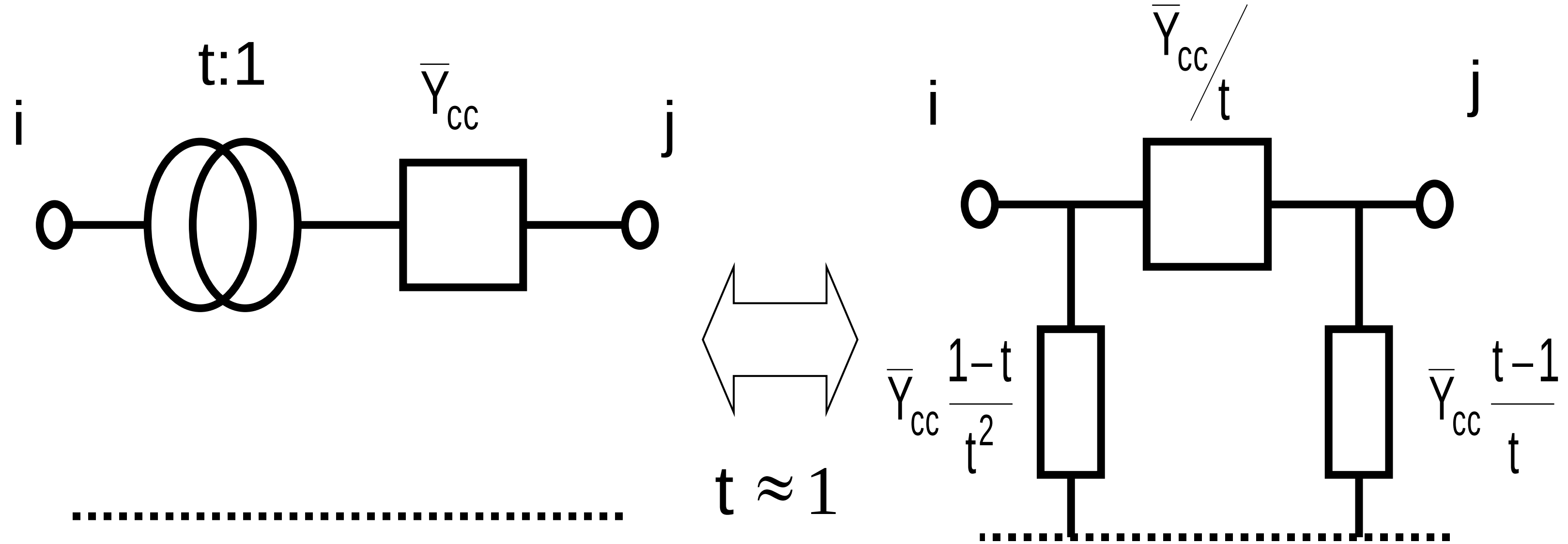
Tap-Changing transformer

- A tap changing transformer makes the admitance matrix dependent on the transformer parameter t.

- The Jacobian matrix also depends on t.

Admitance matrix

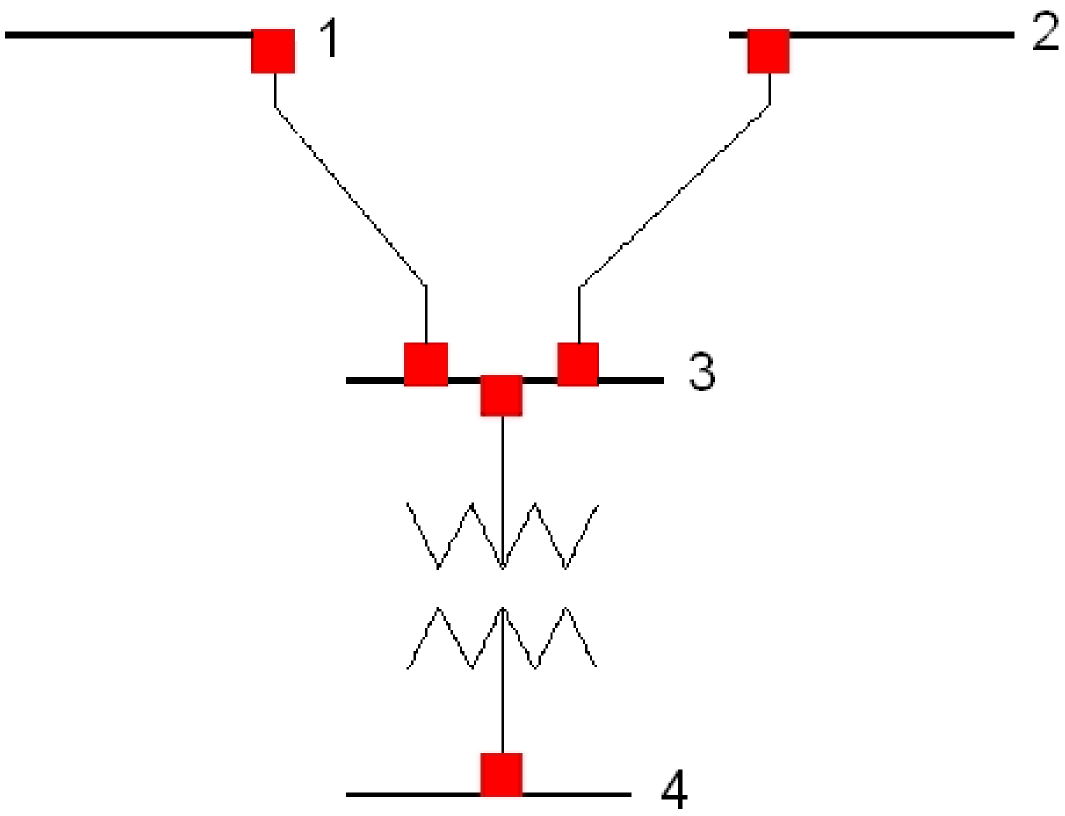
Equivalent circuit:



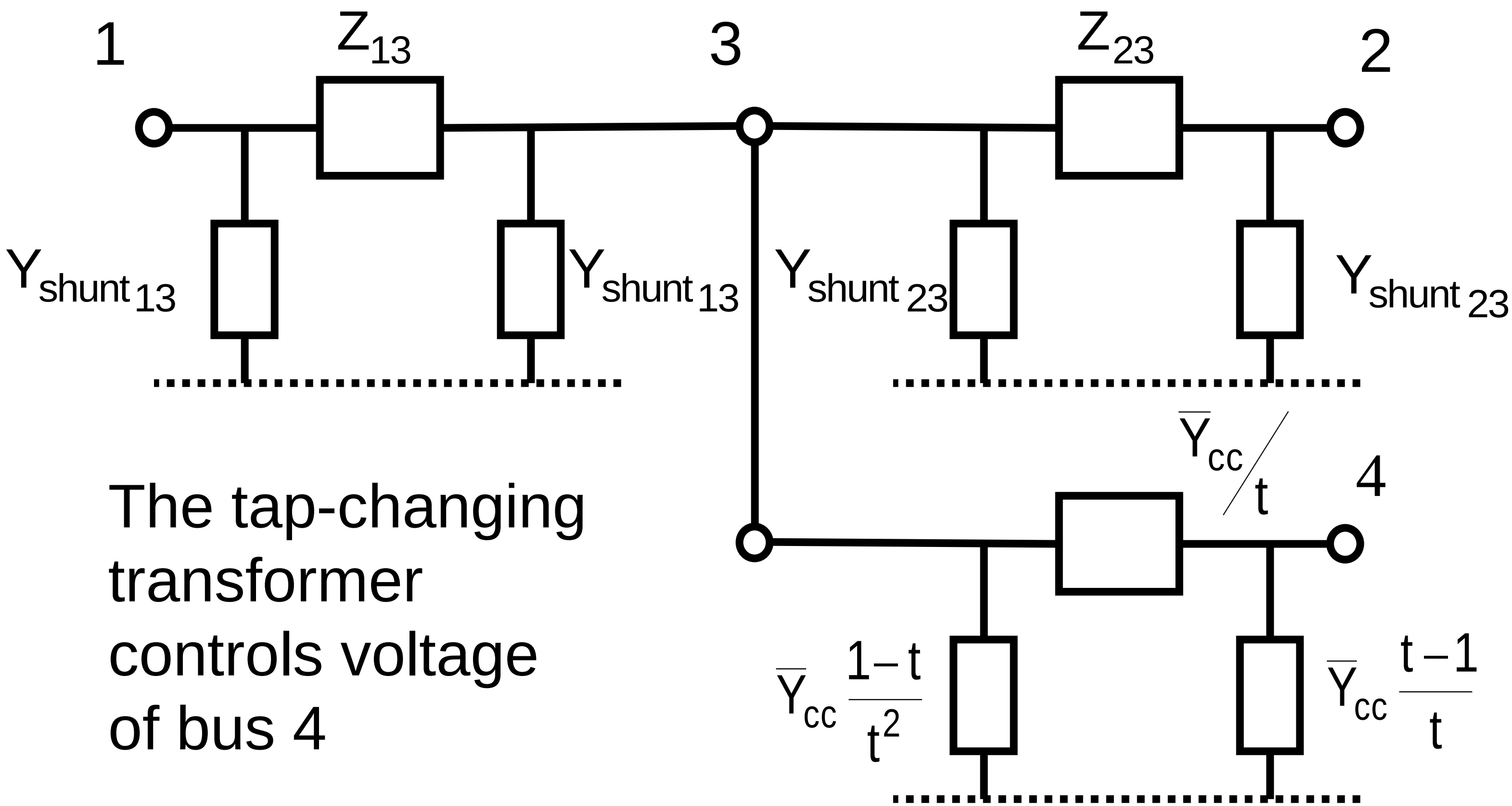
Admitance matrix

- General building rules
  1. Self admittance of node i, $\overline{Y}_{ii}$, equals the algebraic sum of all the admittances connected to node i
  2. Mutual admittance between nodes i and k, $\overline{Y}_{ii}$, equals the negative of the sum of all admittances connecting nodes i and k
  3. $\overline{Y}_{ik} = \overline{Y}_{ki}$

Tap-changing Example



Tap-changing Example



The tap-changing transformer controls voltage of bus 4

Tap-changing Example

$$Y_{11} = \frac{1}{Z_{13}} + Y_{shunt\,13} \;; Y_{12} = 0 \;; \ldots$$

$$\ldots Y_{33} = \frac{1}{Z_{13}} + Y_{shunt\,13} + \frac{1}{Z_{23}} + Y_{shunt\,23} + \overline{Y}_{cc}\frac{1-t}{t^2} + \overline{Y}_{cc}/t$$

$$Y_{34} = -\overline{Y}_{cc}/t$$

$$Y_{44} = \overline{Y}_{cc}/t + \overline{Y}_{cc}\frac{t-1}{t} = \overline{Y}_{cc}$$

It does not depend on t!

Tap-changing Example

$$\overline{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33}(t) & Y_{34}(t) \\ Y_{41} & Y_{42} & Y_{43}(t) & Y_{44} \end{bmatrix}$$

Tap-changing transformer

Load flow equations:

$$P_i = V_i \times \sum_{k=1}^{n} V_k \left[G_{ik} \cos\delta_{ik} + B_{ik} \sin\delta_{ik} \right]$$

$$Q_i = V_i \times \sum_{k=1}^{n} V_k \left[G_{ik} \sin\delta_{ik} - B_{ik} \cos\delta_{ik} \right]$$

Tap-changing transformer

Taylor Expansion:

$$P_i^{dato} \approx f_{iP}^{(r)} + \left[\frac{\partial f_{iP}}{\partial \delta_2}\right]^{(r)} \Delta\delta_2^{(r)} + \ldots + \left[\frac{\partial f_{iP}}{\partial \delta_n}\right]^{(r)} \Delta\delta_n^{(r)} + \left[\frac{\partial f_{iP}}{\partial V_{m+1}}\right]^{(r)} \Delta V_{m+1}^{(r)} + \ldots$$

$$\ldots + \left[\frac{\partial f_{iP}}{\partial V_n}\right]^{(r)} \Delta V_n^{(r)} + \left[\frac{\partial f_{iP}}{\partial t}\right]^{(r)} \Delta t^{(r)}$$

$$Q_i^{dato} \approx f_{iQ}^{(r)} + \left[\frac{\partial f_{iQ}}{\partial \delta_2}\right]^{(r)} \Delta\delta_2^{(r)} + \ldots + \left[\frac{\partial f_{iQ}}{\partial \delta_n}\right]^{(r)} \Delta\delta_n^{(r)} + \left[\frac{\partial f_{iQ}}{\partial V_{m+1}}\right]^{(r)} \Delta V_{m+1}^{(r)} + \ldots$$

$$\ldots + \left[\frac{\partial f_{iQ}}{\partial V_n}\right]^{(r)} \Delta V_n^{(r)} + \left[\frac{\partial f_{iQ}}{\partial t}\right]^{(r)} \Delta t^{(r)}$$

Tap-changing transformer

- The admitance matrix depend on t.
- The Jacobian matrix has a new column.
- The new variable (t) replace the voltage value at the corresponding PQ bus.

Tap-changing transformer

- Increment Δt is divided by t to improve computational efficiency.
- The Jacobian matrix maintains its # of column & rows.
- The variable t is considered in the last place.

Tap-changing transformer

The Jacobian matrix blocks are:

$$J = \begin{bmatrix} H^{(r)} & N^{(r)} & C^{(r)} \\ M^{(r)} & L^{(r)} & D^{(r)} \end{bmatrix}$$

H, N, M & L are the standard blocks.

Submatrix C & D have as many columns as the # of tap-changing transformers.

Tap-changing transformer

Derivatives with respect to t:

$$C_i = t \times \frac{\partial f_{iP}}{\partial t} = t \times V_i \times \sum_{k=1}^{n} V_k \left(\frac{\partial G_{ik}}{\partial t} \cos\delta_{ik} + \frac{\partial B_{ik}}{\partial t} \sin\delta_{ik} \right)$$

$$D_i = t \times \frac{\partial f_{iQ}}{\partial t} = t \times V_i \times \sum_{k=1}^{n} V_k \left(\frac{\partial G_{ik}}{\partial t} \sin\delta_{ik} \qquad \frac{\partial B_{ik}}{\partial t} \cos\delta_{ik} \right)$$

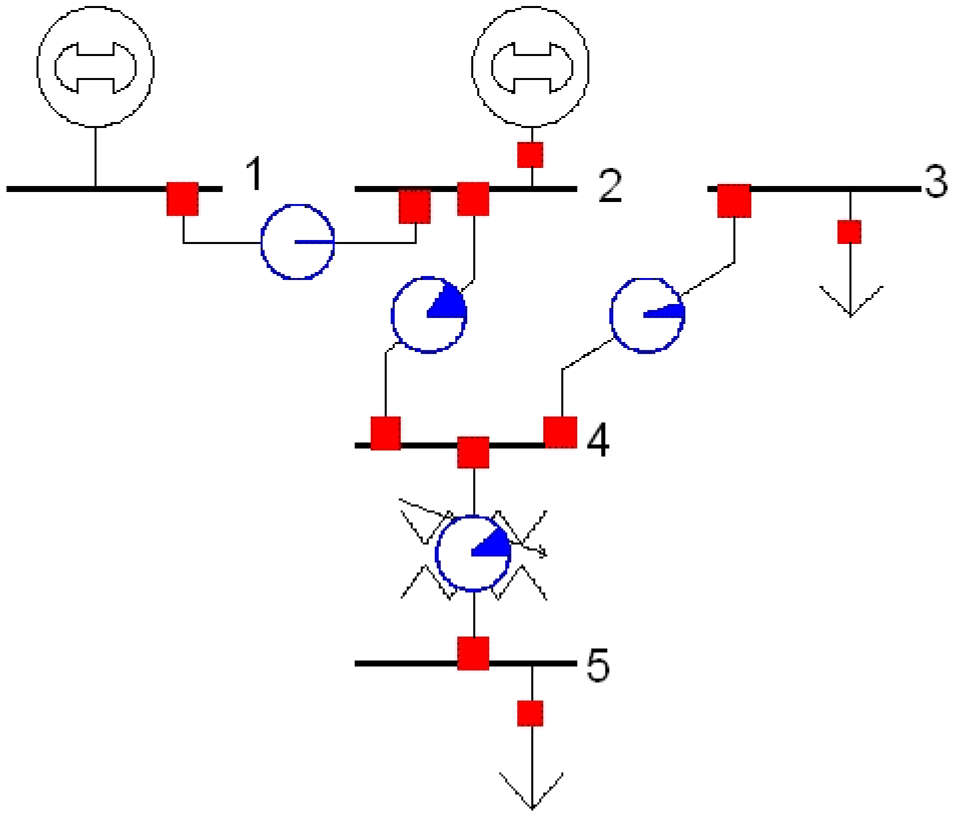
C & D multiplied by t!

Tap-changing transformer

Iterative procedure analogous, but:

- If t hit any of its limit, it is fixed to the limit & the corresponding bus becomes PQ.
- If the procedure converge, fix t at its closest integer value & continues the iteration with that t fixed.

Tap-changing Example



1.- Slack
2.- PV
3.- PQ
4.- PQ
5.- PQV

Tap-changing Example

Data:

Line impedances: 0.001+j0.05 p.u.

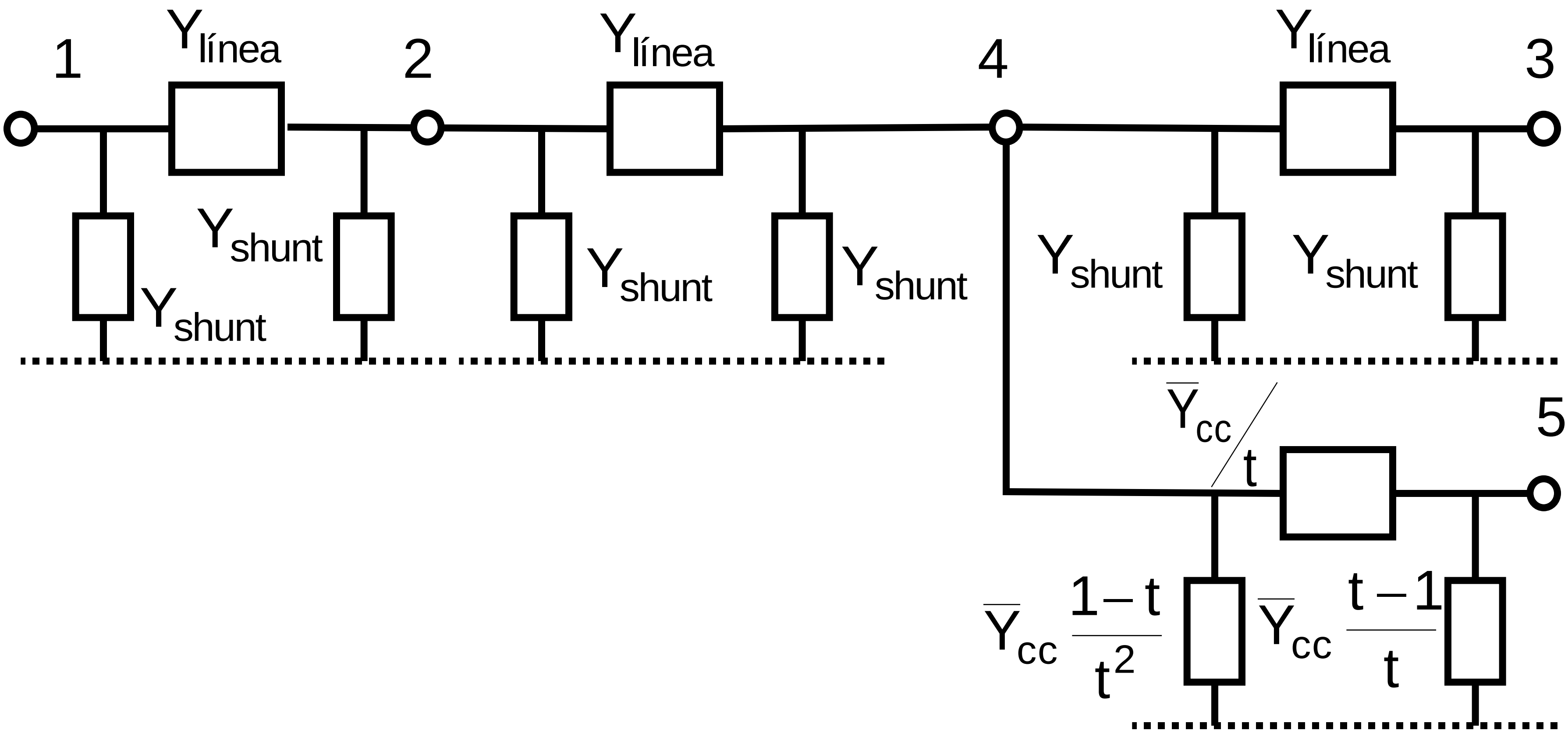
Shunt admitances: j0.05 p.u. (2 per line).

Transformer: $0.9<t<1.1$, steps of 0.005; Z_{cc}=j0.1.

Tap-changing Example

Bus	Voltage (pu).	Power
1	1.00	Slack
2	1.00	P_g=150MW
3	--	P_c=50MW, Q_c=10MVAr
4	--	P_c=0MW, Q_c=0MVAr
5	1.00	P_c=100MW,Q_c=50MVAr

Tap-changing Example



Tap-changing Example

Admitance matrix:

$$\overline{Y}_{BUS} = \begin{bmatrix} Y_l + Y_s & Y_l & 0 & 0 & 0 \\ Y_l & 2Y_l + 2Y_s & 0 & Y_l & 0 \\ 0 & 0 & Y_l + Y_s & Y_l & 0 \\ 0 & Y_l & Y_l & 2Y_l + 2Y_s + \dfrac{Y_{cc}}{t} + Y_{cc}\dfrac{(1 \quad t)}{t^2} & \dfrac{Y_{cc}}{t} \\ 0 & 0 & 0 & \dfrac{Y_{cc}}{t} & Y_{cc} \end{bmatrix}$$

Tap-changing Example

$$\overline{Y}_{bus} = \begin{bmatrix} 0.3998 - j19.942 & -0.3998 + j19.992 & 0 & 0 & 0 \\ -0.3998 + j19.992 & 0.7997 - j39.884 & 0 & -0.3998 + j19.992 & 0 \\ 0 & 0 & 0.3998 - j19.942 & -0.3998 + j19.992 & 0 \\ 0 & -0.3998 + j19.992 & -0.3998 + j19.992 & 0.7997 - j49.884 & j10 \\ 0 & 0 & 0 & j10 & -j10 \end{bmatrix}$$

Variable initial values:

$$\delta_2 = \delta_3 = \delta_4 = \delta_5 = 0;$$

$$V_3 = V_4 = 1;$$

$$t = 1;$$

Tap-changing Example

Increment calculations:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta P_5 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \end{bmatrix} = \begin{bmatrix} 1.5 - \Delta P_2^{cal} \\ -0.5 - \Delta P_3^{cal} \\ -\Delta P_4^{cal} \\ -1 - \Delta P_5^{cal} \\ -0.1 - \Delta Q_3^{cal} \\ -\Delta Q_4^{cal} \\ -0.5 - \Delta Q_5^{cal} \end{bmatrix} = \begin{bmatrix} 1.5 \\ -0.5 \\ 0 \\ -1 \\ -0.1 \\ 0 \\ -0.5 \end{bmatrix}$$

Tap-changing Example

Jacobian

$$\bar{J} = \begin{bmatrix} 39.984 & 0 & -19.992 & 0 & 0 & -0.3998 & 0 \\ 0 & 19.992 & -19.992 & 0 & 0.3998 & -0.3998 & 0 \\ -19.992 & -19.992 & 49.984 & -10 & -0.3998 & 0.7997 & 0 \\ 0 & 0 & -10 & 10 & 0 & 0 & 0 \\ 0 & -0.3998 & 0.3998 & 0 & 19.892 & -19.992 & 0 \\ 0.3998 & 0.3998 & -0.7997 & 0 & -19.992 & 49.754 & -10 \\ 0 & 0 & 0 & 0 & 0 & -10 & 10 \end{bmatrix}$$

Tap-changing Example

First iteration:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta\delta_5 \\ \frac{\Delta V_3}{V_3} \\ \frac{\Delta V_4}{V_4} \\ \frac{\Delta t}{t} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.0993 \\ -0.0744 \\ -0.1744 \\ -0.0377 \\ -0.032 \\ -0.082 \end{bmatrix} \Rightarrow V = \begin{bmatrix} 1 \\ 1 \\ 0.9623 \\ 0.968 \\ 1 \end{bmatrix}; \delta = \begin{bmatrix} 0 \\ 0 \\ -0.0993 \\ -0.0744 \\ -0.1744 \end{bmatrix}; t = 0.918$$

Tap-changing Example

Admitance matrix:

$$
\overline{Y}_{bus} = \begin{bmatrix} 0.3998 - j19.942 & -0.3998 + j19.99 & 0 & 0 & 0 \\ -0.3998 + j19.992 & 0.7997 - j39.884 & 0 & -0.3998 + j19.992 & 0 \\ 0 & 0 & 0.3998 - j19.94 & -0.3998 + j19.992 & 0 \\ 0 & -0.3998 + j19.992 & -0.399 + j19.99 & 0.7997 - j51.750 & j10.8933 \\ 0 & 0 & 0 & j10.893 & -j10 \end{bmatrix}
$$

Tap-changing Example

Power computation:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.4521 \\ -0.4657 \\ 0.069 \\ -1.0527 \end{bmatrix} \qquad \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} -0.05 \\ 0.5647 \\ -0.1407 \\ 0.1027 \\ -0.492 \end{bmatrix}$$

Tap-changing Example

Power increments:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta P_5 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \end{bmatrix} = \begin{bmatrix} 1.5 - 1.4521 \\ -0.5 + 0.4657 \\ -0.069 \\ -1 + 1.0527 \\ -0.1 + 0.1407 \\ -0.1027 \\ -0.5 + 0.492 \end{bmatrix} = \begin{bmatrix} 0.0479 \\ -0.0343 \\ -0.069 \\ 0.0527 \\ 0.0407 \\ -0.1027 \\ -0.008 \end{bmatrix}$$

Tap-changing Example

Jacobian matrix:

$$\bar{J} = \begin{bmatrix} 39.9193 & 0 & -19.3273 & 0 & 0 & 1.0523 & 0 \\ 0 & 18.6075 & -18.6075 & 0 & -0.0954 & -0.8359 & 0 \\ -19.2698 & -18.6261 & 48.3879 & -10.492 & 0.0912 & 0.8183 & -1.0527 \\ 0 & 0 & -10.492 & 10.492 & 0 & -1.0527 & 1.0527 \\ 0 & -0.8359 & 0.8359 & 0 & 18.3261 & -18.6075 & 0 \\ 1.8242 & -0.0912 & -0.6803 & -1.0527 & -18.6261 & 48.5934 & -11.746 \\ 0 & 0 & 1.0527 & -1.0527 & 0 & -10.492 & 10.492 \end{bmatrix}$$

Tap-changing Example

Second iteration

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta\delta_5 \\ \frac{\Delta V_3}{V_3} \\ \frac{\Delta V_4}{V_4} \\ \frac{\Delta t}{t} \end{bmatrix} = \begin{bmatrix} -0.0002 \\ -0.005 \\ -0.003 \\ 0.0021 \\ -0.0016 \\ -0.0037 \\ -0.0039 \end{bmatrix} \Rightarrow V = \begin{bmatrix} 1 \\ 1 \\ 0.9607 \\ 0.9644 \\ 1 \end{bmatrix}; \delta = \begin{bmatrix} 0 \\ -0.0002 \\ -0.1043 \\ -0.0744 \\ -0.1723 \end{bmatrix}; t = 0.9144$$

Tap-changing Example

Admitance matrix:

$$\overline{Y}_{bus} = \begin{bmatrix} 0.3998 - j19.942 & -0.3998 + j19.992 & 0 & 0 & 0 \\ -0.3998 + j19.992 & 0.7997 - j39.884 & 0 & -0.3998 + j19.992 & 0 \\ 0 & 0 & 0.3998 - j19.942 & -0.3998 + j19.992 & 0 \\ 0 & -0.3998 + j19.992 & -0.3998 + j19.992 & 0.7997 - j51.8447 & j10.9365 \\ 0 & 0 & 0 & j10.9365 & -j10 \end{bmatrix}$$

Tap-changing Example

Power calculations:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0.0031 \\ 1.4998 \\ -0.4998 \\ 0.000 \\ -1.0000 \end{bmatrix} \qquad \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} -0.0501 \\ 0.6391 \\ -0.1000 \\ 0.0006 \\ -0.4999 \end{bmatrix}$$

Tap-changing Example

Power increments:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta P_5 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \end{bmatrix} = \begin{bmatrix} 1.5 - 1.4998 \\ -0.5 + 0.4998 \\ 0.000 \\ -1 + 1.0000 \\ -0.1 + 0.1000 \\ -0.0006 \\ -0.5 + 0.4999 \end{bmatrix} = \begin{bmatrix} 0.2065 \\ -0.1969 \\ 0.0057 \\ -0.0258 \\ 0.0339 \\ -0.6375 \\ -0.1218 \end{bmatrix} \times 10^{-3}$$

Tap-changing Example

Tap to the closest feasible value:

$$t = 0.9144 \Rightarrow t = 0.915$$

Admitance matrix:

$$\bar{Y}_{bus} = \begin{bmatrix} 0.3998 - j19.942 & -0.3998 + j19.992 & 0 & 0 & 0 \\ -0.3998 + j19.992 & 0.7997 - j39.884 & 0 & -0.3998 + j19.992 & 0 \\ 0 & 0 & 0.3998 - j19.942 & -0.3998 + j19.992 & 0 \\ 0 & -0.3998 + j19.992 & -0.3998 + j19.992 & 0.7997 - j51.8282 & j10.929 \\ 0 & 0 & 0 & j10.929 & -j10 \end{bmatrix}$$

Tap-changing Example

Power calculation:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0.0031 \\ 1.4998 \\ -0.4998 \\ 0.0007 \\ -0.9993 \end{bmatrix} \qquad \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} -0.0501 \\ 0.6391 \\ -0.1000 \\ -0.0075 \\ -0.4926 \end{bmatrix}$$

Tap-changing Example

Power increment calculations:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta P_5 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \end{bmatrix} = \begin{bmatrix} 1.5 - 1.4998 \\ -0.5 + 0.4998 \\ -0.0007 \\ -1 + 0.9993 \\ -0.1 + 0.1000 \\ 0.0075 \\ -0.5 + 0.4926 \end{bmatrix} = \begin{bmatrix} 0.0002 \\ -0.0002 \\ 0.0007 \\ -0.0007 \\ 0.0000 \\ 0.0075 \\ -0.0074 \end{bmatrix} ; \begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta\delta_5 \\ \frac{\Delta V_3}{V_3} \\ \frac{\Delta V_4}{V_4} \\ \frac{\Delta V_5}{V_5} \end{bmatrix}$$

Tap-changing Example

Jacobian:

$$\bar{J} = \begin{bmatrix} 39.2449 & 0 & -19.253 & 0 & 0 & 1.103 & 0 \\ 0 & 18.5072 & -18.5072 & 0 & -0.1307 & -0.8689 & 0 \\ -19.1935 & -18.5271 & 48.2132 & -10.4926 & 0.1282 & 0.7431 & 0.9993 \\ 0 & 0 & -10.4926 & 10.4926 & 0 & -0.9993 & -0.9993 \\ 0 & -0.8689 & 0.8689 & 0 & 18.3071 & -18.5072 & 0 \\ 1.872 & -0.1282 & -0.7445 & -0.9993 & -18.5271 & 48.1983 & -10.4926 \\ 0 & 0 & 0.9993 & -0.9993 & 0 & -10.4926 & 9.5074 \end{bmatrix}$$

Tap-changing Example

Final values

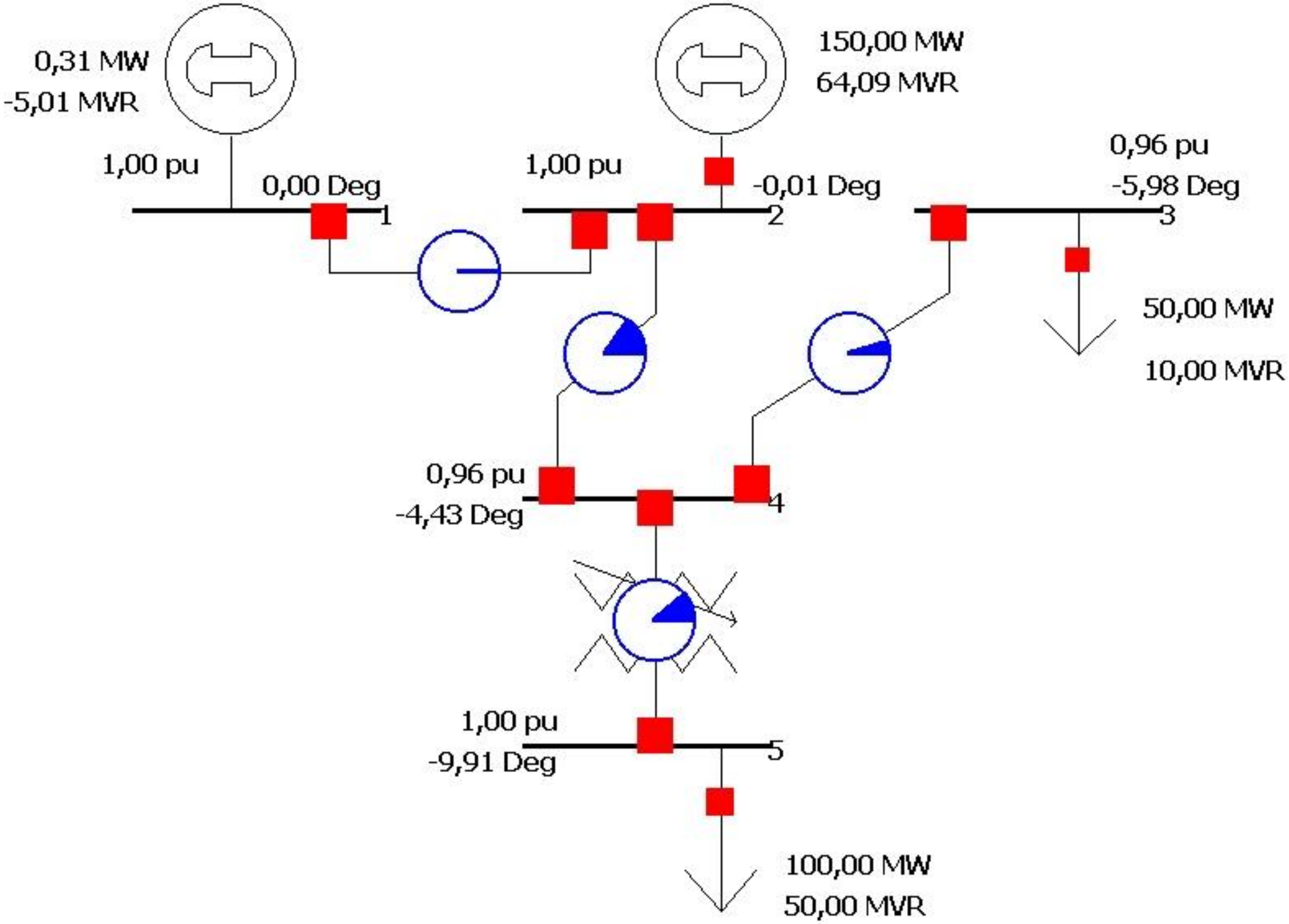
$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta\delta_5 \\ \frac{\Delta V_3}{V_3} \\ \frac{\Delta V_4}{V_4} \\ \frac{\Delta V_5}{V_5} \end{bmatrix} = \begin{bmatrix} -0.0007 \\ -0.0289 \\ -0.0152 \\ -0.1699 \\ -0.0552 \\ -0.0558 \\ -0.8522 \end{bmatrix} \times 10^{-3} \Rightarrow V = \begin{bmatrix} 1 \\ 1 \\ 0.9607 \\ 0.9644 \\ 0.9991 \end{bmatrix}; \delta = \begin{bmatrix} 0 \\ -0.0002 \\ -0.1043 \\ -0.0744 \\ -0.1725 \end{bmatrix}$$

Tap-changing Example

Final power values:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0.0031 \\ 1.5000 \\ -0.5000 \\ 0.0000 \\ -1.0000 \end{bmatrix} \qquad \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} -0.0501 \\ 0.6402 \\ -0.1000 \\ 0.000 \\ -0.5000 \end{bmatrix}$$

Power World



0,31 MW
-5,01 MVR
150,00 MW
64,09 MVR
1,00 pu
0,00 Deg
1
1,00 pu
-0,01 Deg
2
0,96 pu
-5,98 Deg
3
50,00 MW
10,00 MVR
0,96 pu
-4,43 Deg
4
1,00 pu
-9,91 Deg
5
100,00 MW
50,00 MVR

5. Fast decoupled AC load flow

Fast decoupled AC load flow

Two simplifications:

- Do not build Jacobian at each iteration (small error introduced, then, the procedure needs more iterations to reach the solution)

- Decoupling between P-δ and Q-V (not recommended in system highly loaded and/or with low voltage levels)

Fast decoupled AC load flow

$$\begin{bmatrix} \Delta P^{(r)} \\ \Delta Q^{(r)} \end{bmatrix} = \begin{bmatrix} H^{(r)} & N^{(r)} \\ J^{(r)} & L^{(r)} \end{bmatrix} \begin{bmatrix} \Delta\delta^{(r+1)} \\ \Delta V^{(r+1)} / V^{(r)} \end{bmatrix}$$

Assume:

i) $\left|\overline{V}_i\right| \approx 1.0 \quad \forall i$

ii) $G_{ik} << B_{ik} \quad \forall i,k$

iii) $\cos(\delta_i - \delta_k) \approx 1.0$

$\sin(\delta_i - \delta_k) \approx 0.0$

iv) $\left|Q_k\right| << B_{kk}$

Fast decoupled AC load flow

We have:

$$[H] = [\widetilde{B}_1] \ , \quad [L] = [\widetilde{B}_2], \quad [N] = [0], \quad [J] = [0]$$

$$[\widetilde{B}_1],[\widetilde{B}_2] = \begin{Bmatrix} \widetilde{B}_{ik} = -B_{ik} \\ \widetilde{B}_{ii} = -B_{ii} \end{Bmatrix} \quad \rightarrow \quad \text{Elements of } \overline{Y}_{BUS}$$

Fast decoupled AC load flow

$$\begin{bmatrix} \Delta P^{(r)} \\ \Delta Q^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{B}_1 & 0 \\ 0 & \tilde{B}_2 \end{bmatrix} \begin{bmatrix} \Delta\delta^{(r+1)} \\ \Delta V^{(r+1)} \end{bmatrix}$$

$$\Delta P^{(r)} = \left[\tilde{B}_1\right]\left[\Delta\delta^{(r+1)}\right] \qquad (1)$$

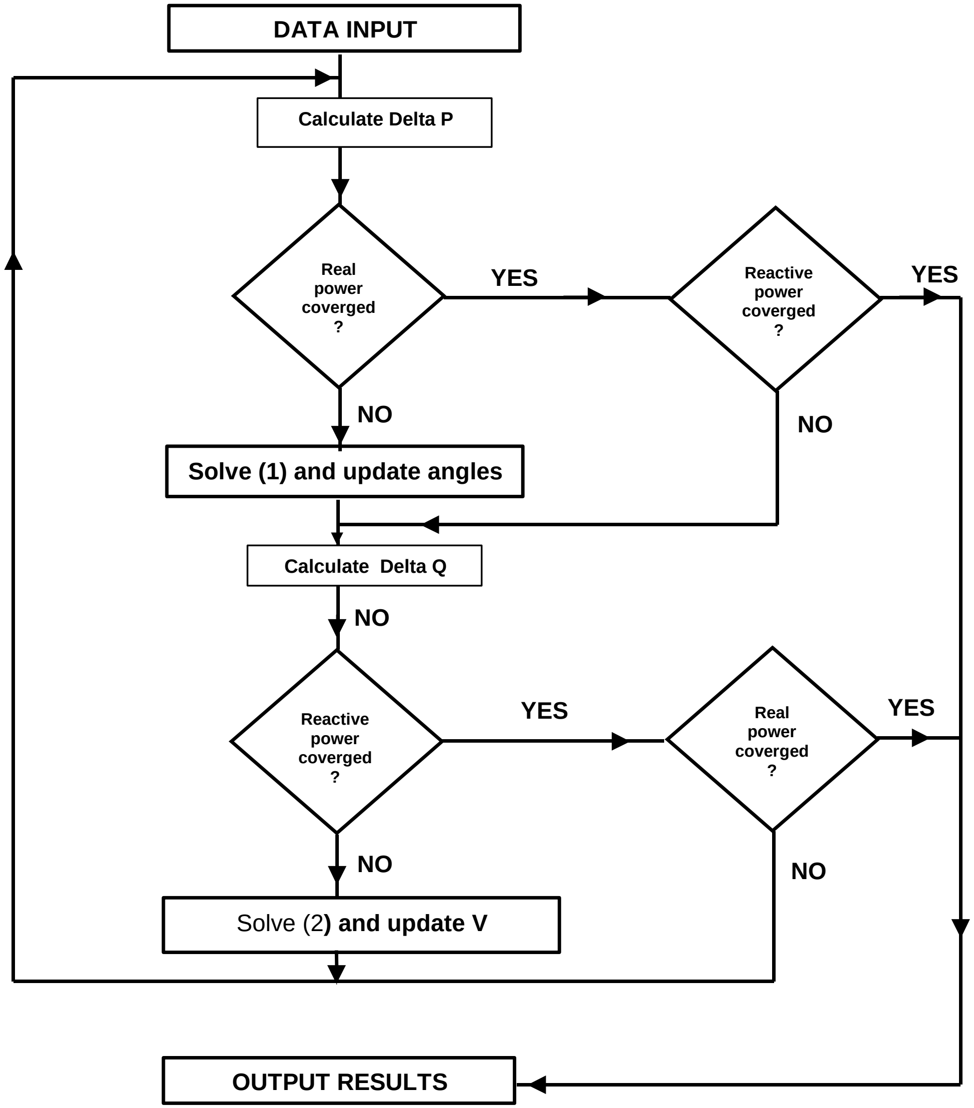
$$\Delta Q^{(r)} = \left[\tilde{B}_2\right]\left[\Delta V^{(r+1)}\right] \qquad (2)$$

$$\left[\Delta\delta^{(r+1)}\right] = \left[\tilde{B}_1\right]^{-1}\left[\Delta P^{(r)}\right]$$

$$\left[\Delta V^{(r+1)}\right] = \left[\tilde{B}_2\right]^{-1}\left[\Delta Q^{(r)}\right]$$

Use Newton-Raphson Iteration

Fast decoupled load flow. Flow diagram



Fast decoupled load flow. Matlab Code

```
function [V_modulo, V_fase, P, Q, N_iter, T_calculo, Error_p] = desacoplado(V_modulo_ini, V_fase_ini, P_ini, Q_ini, Y,
        npv)

%
% [V_modulo, V_fase, P, Q, N_iter, T_calculo, Error_p] = desacoplado(V_modulo_ini, V_fase_ini, P_ini, Q_ini, Y, npv)
%
% Obtencion de indices de nodos:    1          ->  SLACK;
%                                   2,...,M     ->  PV;
%                                   M+1,...,N   ->  PQ

n = length(V_modulo_ini);
m = npv + 1;
% Parametros del Metodo
Tol     = 0.0001;                         % Tolerancia del metodo (Perdida de potencia).
Error_p = 1;                              % Error inicial (A un valor mayor que Tol).
N_iter  = 0;                              % Numero de iteraciones.
% Valores iniciales
V_modulo= V_modulo_ini';
V_fase  = V_fase_ini';
V       = V_modulo.*exp(j*V_fase);        % Expresion compleja de la tension
P       = P_ini';
Q       = Q_ini';

DS_DA = diag(V)*conj(Y*j*diag(V)) + diag(conj(Y*V))*j*diag(V);
DS_DV = diag(V)*conj(Y*diag(V./V_modulo)) + diag(conj(Y*V))*diag(V./V_modulo);
```

Fast decoupled load flow. Matlab Code

```
%J    = [real(DS_DA(2:n   , 2:n)) real(DS_DV(2:n   , m+1:n))      % Construccion del Jacobiano
%        imag(DS_DA(m+1:n , 2:n)) imag(DS_DV(m+1:n , m+1:n))];

H    = imag(-Y(2:end  , 2:end))
L    = imag(-Y(m+1:end,m+1:end))

j=sqrt(-1);
tic;
while (Error_p > Tol)                          % Bucle principal  (Se permiten 50 iteraciones como mucho).
   if (N_iter >50)
      error('Demasiadas iteraciones');
      break
   end

   V     = V_modulo.*exp(j*V_fase);            % Expresion compleja de la tension
   S     = V.*conj(Y*V);                       % Expresion compleja de la potencia
   DP    = P(2:n)-real(S(2:n));                % Incremento de potencia activa (nudos PV y PQ)
   DQ    = Q(m+1:n)-imag(S(m+1:n));            % Incremento de potencia reactiva (nudos PQ)

   Dfase   = H\DP;
   Dmodulo = L\DQ;

   PQ    = [DP ; DQ];
   Error_p = norm(PQ,2);                       % Error en esa iteracion
```

Fast decoupled load flow. Matlab Code

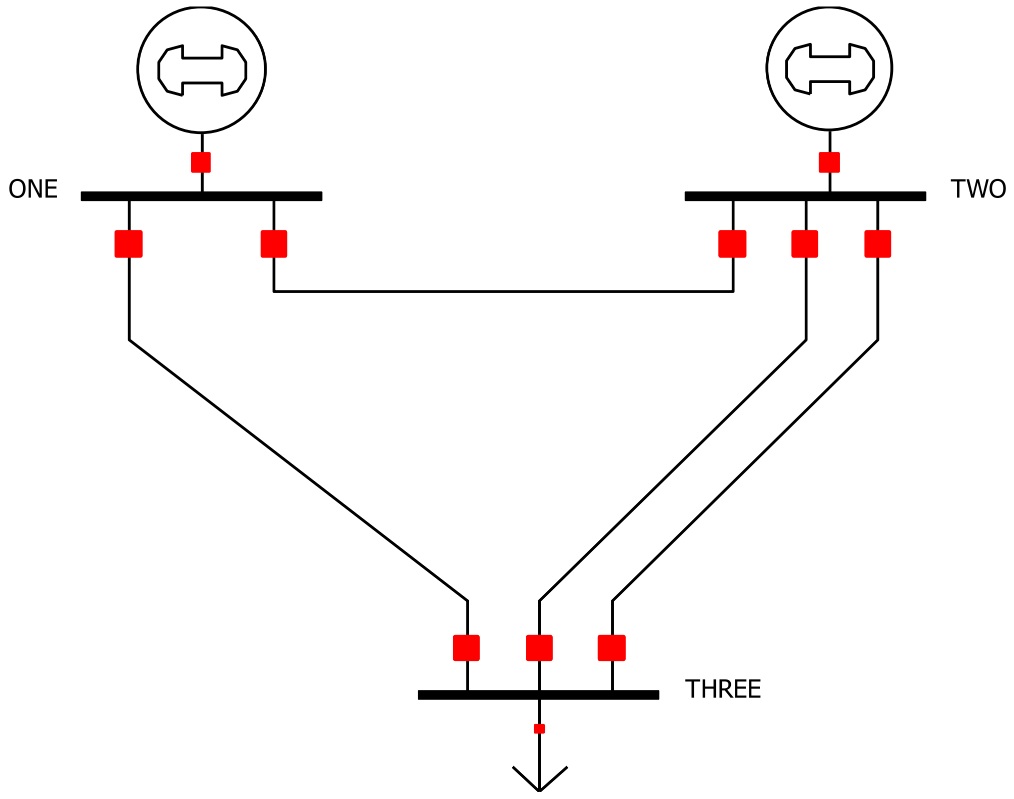
```
   V_fase   (2:n)  = V_fase(2:n)     + Dfase;              % Actualizamos la fase de las tensiones (nudos PV y PQ)
   V_modulo (m+1:n)= V_modulo(m+1:n) + Dmodulo;   % Actualizamos el modulo de las tensiones (nudos PQ)


   N_iter  = N_iter + 1;                                      % Incremento el numero de iteraciones

%     disp('Pulse una tecla para continuar')
%     pause
end

P=real(S);                                                    % Calculo de la potencia activa
Q=imag(S);                                                    % Calculo de la potencia reactiva
V_fase=V_fase*180/pi;                                         % Paso de Radianes a grados
T_calculo=toc;


%               **** ENTRADAS ****
%
%  V_modulo_ini = Modulo de la tension para comenzar a iterar (conocidos en SLACK y PV).
%  V_fase_ini       = Fase de la tension para comenzar a iterar (conocido en SLACK).
%  P_ini                = Potencia activa en los nodos (conocido en PV y PQ).
%  Q_ini                = Potencia reactica en los nodos (conocido en PQ).
%  Y                    = Matriz de admitancias nodales.
```

Fast decoupled load flow. Matlab Code

```
%              **** SALIDAS  ****
%
%  V_modulo     = Modulo de la tension en todos los nodos.
%  V_fase       = Fase de la tension en todos los nodos.
%  P            = Potencia activa en todos los nodos.
%  Q            = Potencia reactiva en todos los nodos.
%  N_iter       = Numero de iteraciones.
%  T_calculo    = Tiempo de calculo.
%  Error_p      = Error.
%
%
%              **** OBSERVACIONES ****
%
%  Los nodos deben estar ordenador asi:
%
%       * 1      : SLACK.
%     * 2...m   : PV.
%   * m+1...n : PQ.
%
%
%              **** FECHA Y AUTOR ****
%
%  Laura Laguna - Nobiembre de 2005
```

Fast decoupled load flow: Exaple

- Tolerance 0.1 MVA.



Fast decoupled load flow: Exaple

- Power base S_B = 100MVA

Bus	Voltage p.u.	Power
1	1.02	-
2	1.02	P_G=50 MW
3	-	P_C=100 MW Q_C=60 MVAr

Line	Impedance p.u.
1-2	0.02+0.04j
1-3	0.02+0.06j
2-3	0.02+0.04j (cada una)

Fast decoupled load flow: Exaple

- Admitance matrix

$$\overline{Y}_{bus}=\begin{bmatrix} 15-35j & -10+20j & -5+15j \\ -10+20j & 30-60j & -20+40j \\ -5+15j & -20+40j & 25-55j \end{bmatrix}$$

Fast decoupled load flow: Exaple

- Data and unknown:

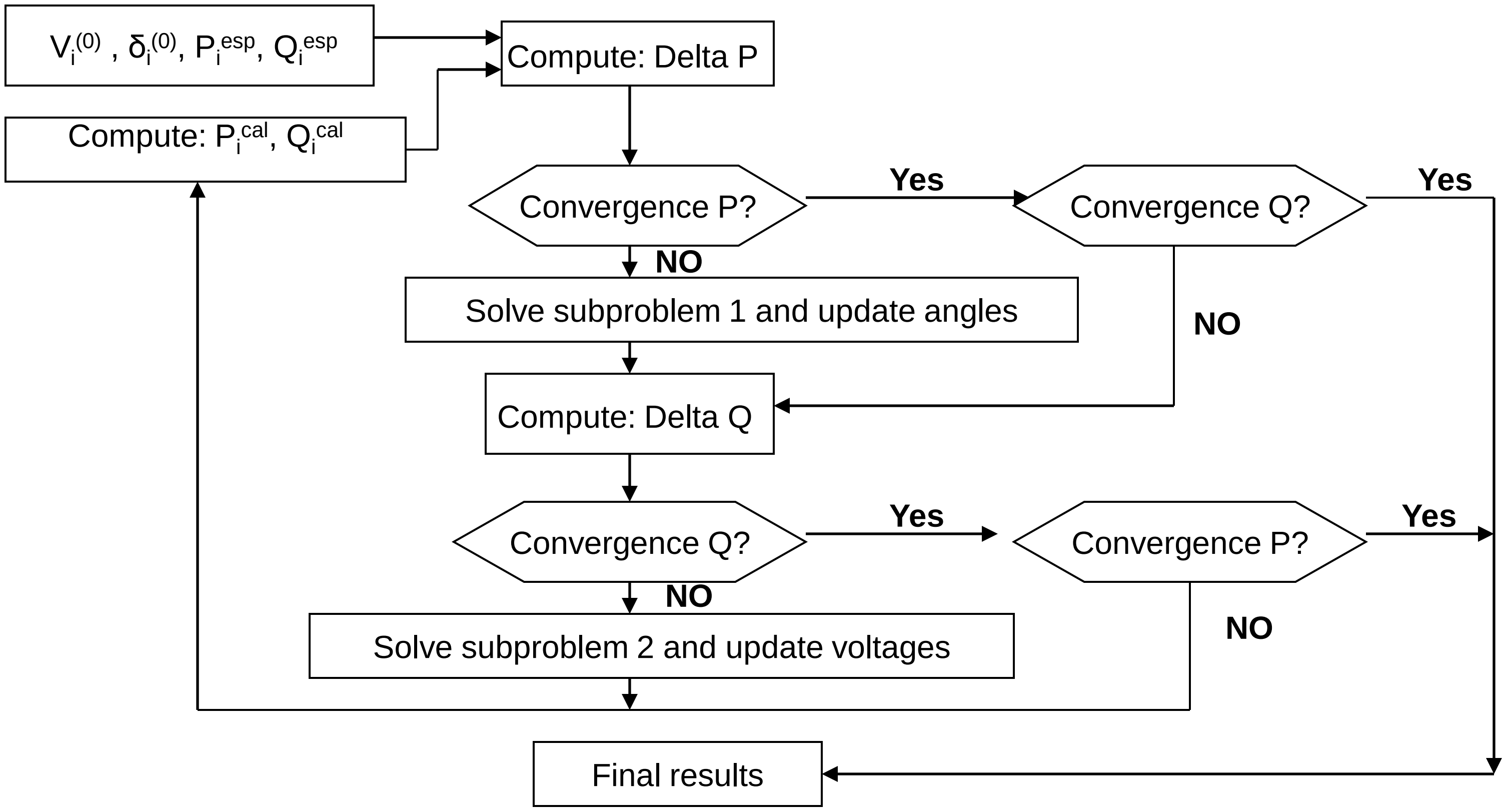
Bus	Type	Data	Unknown
1	Slack and reference	$V_1 = 1.02$ $\delta_1 = 0.0$	P_1, Q_1
2	PV	$P_2 = 0.5$ $V_2 = 1.02$	δ_2, Q_2
3	PQ	$P_3 = -1.0$ $Q_3 = -0.6$	δ_3, V_3

- Inicialization:

$\delta_2 = \delta_3 = 0.0$; $V_3 = 1.02$ p.u.

Fast decoupled load flow: Exaple

- Flow diagram



Fast decoupled load flow: Exaple

Compute $$H = \begin{pmatrix} 60 & -40 \\ -40 & 55 \end{pmatrix}; \quad L = (55)$$

Calculate P and Q:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.3624 \cdot 10^{-14} \\ 0 \\ 7.24755 \cdot 10^{-15} \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} 0.5 \\ -1 \\ -0.6 \end{bmatrix}$$

No convergence

Fast decoupled load flow: Exaple

- First iteration:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.4213 \\ -1.3481 \\ -0.0109 \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0091 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4213 \\ -1.3481 \end{pmatrix}$$

Fast decoupled load flow: Exaple

Residuals:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.7385 \\ -1.3003 \\ -0.1417 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} -0.2385 \\ 0.3003 \\ -0.4583 \end{bmatrix}$$

Error = 0.5976 → no convergence

Fast decoupled load flow: Exaple

- System state (second iteration)

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.0372 \\ 0.2857 \\ -0.0083 \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0008 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4585 \\ -1.0264 \end{pmatrix}$$

Fast decoupled load flow: Exaple

- Residuals:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.6578 \\ -1.1936 \\ -0.7444 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} -0.1578 \\ 0.1936 \\ 0.1444 \end{bmatrix}$$

- Error = 0.2885 $\rightarrow$ no convergence

Fast decoupled load flow: Exaple

- System state (third iteration):

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.0315 \\ 0.1788 \\ 0.0026 \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0034 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4900 \\ -0.8836 \end{pmatrix}$$

- The process continues.

Fast decoupled load flow: Exaple

- After 9 iterations:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.5006 \\ -1.0007 \\ -0.5992 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} -0.5954 \cdot 10^{-3} \\ 0.7006 \cdot 10^{-3} \\ -8.3346 \cdot 10^{-3} \end{bmatrix}$$

- Error = 0.0012 $\rightarrow$ convergence attained

Fast decoupled load flow: Exaple

- System state at iteration 10:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.1592 \cdot 10^{-3} \\ 0.6141 \cdot 10^{-3} \\ -1.5154 \cdot 10^{-5} \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0043 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4710 \\ -0.9614 \end{pmatrix}$$

Fast decoupled load flow: Exaple

- Residuals:

$$\begin{bmatrix} P_2^{cal} \\ P_3^{cal} \\ Q_3^{cal} \end{bmatrix} = \begin{bmatrix} 0.5003 \\ -1.0004 \\ -0.6003 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} 0.5 - P_2^{cal} \\ -1 - P_3^{cal} \\ -0.6 - Q_3^{cal} \end{bmatrix} \approx \begin{bmatrix} -0.2863 \cdot 10^{-3} \\ 0.3516 \cdot 10^{-3} \\ 3.3349 \cdot 10^{-4} \end{bmatrix}$$

- Error = $5.6285 \cdot 10^{-4} \rightarrow$ convergence attained

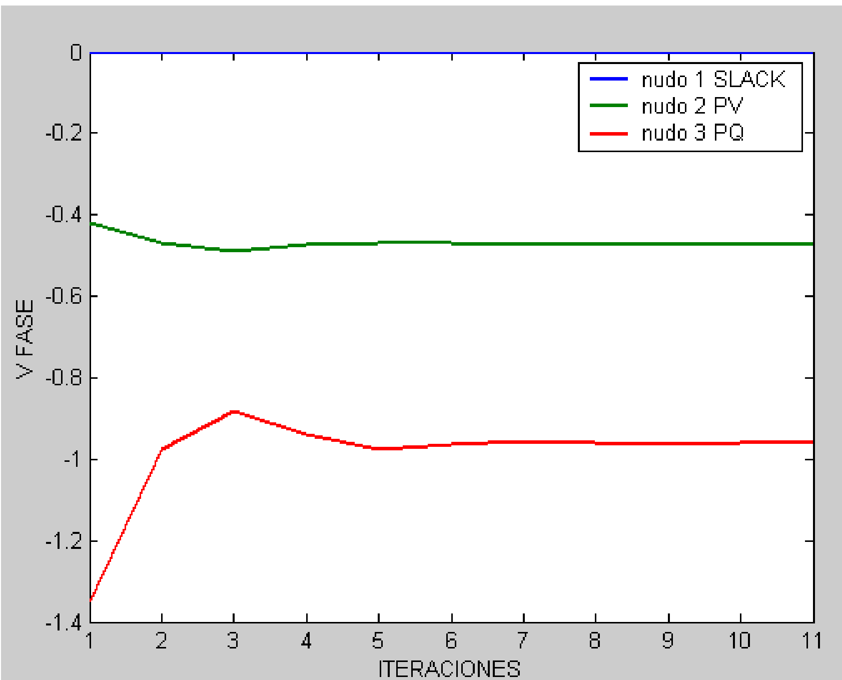
Fast decoupled load flow: Exaple

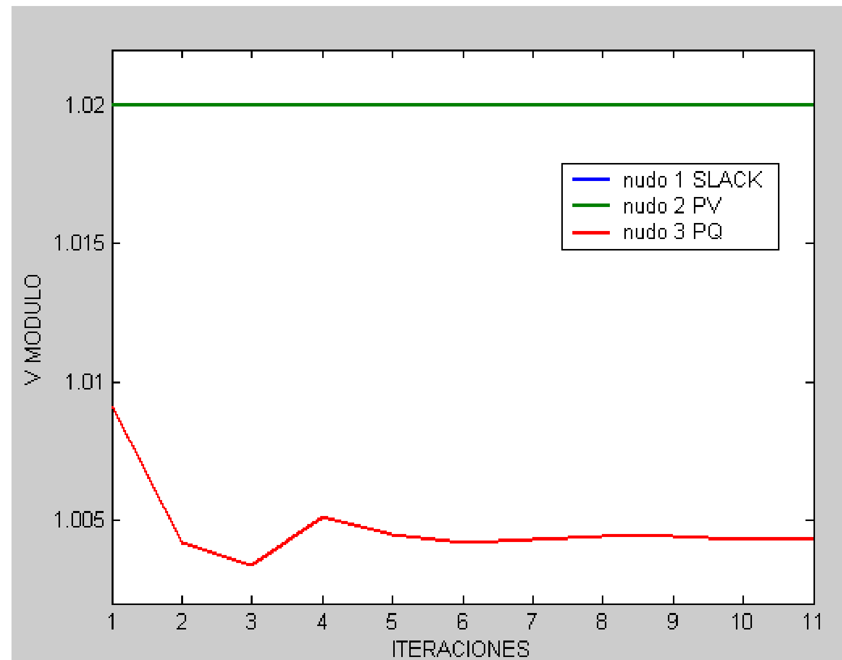
- State at iteration 11:

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta V_3 / V_3 \end{bmatrix} = \begin{bmatrix} -0.0567\cdot 10^{3} \\ 0.3251\cdot 10^{3} \\ 6.0634\cdot 10^{6} \end{bmatrix} \longrightarrow V = \begin{pmatrix} 1.0200 \\ 1.0200 \\ 1.0043 \end{pmatrix}; \quad \delta = \begin{pmatrix} 0 \\ -0.4711 \\ -0.9611 \end{pmatrix}$$

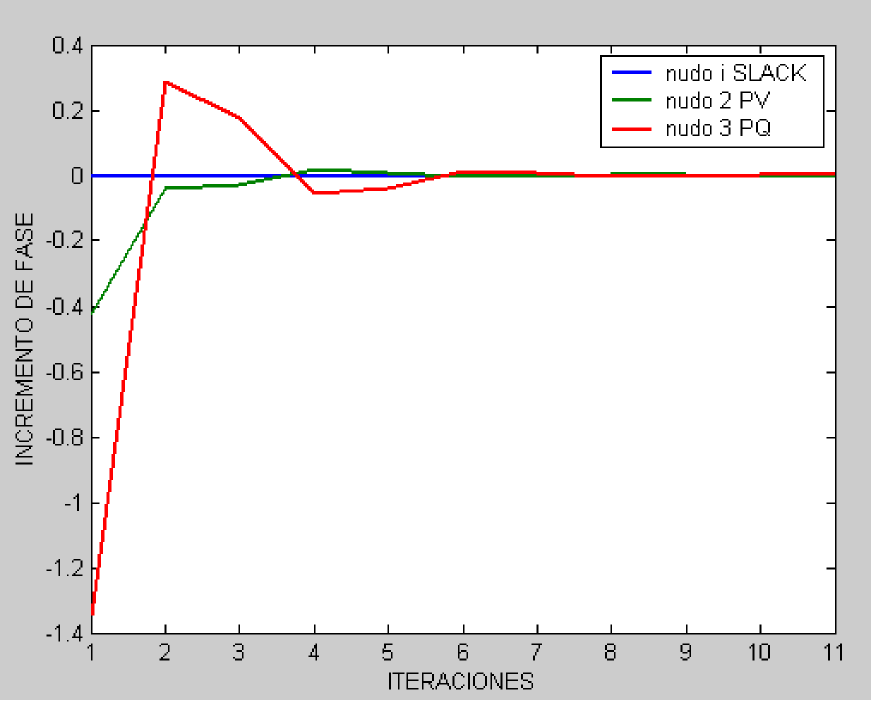
$$P = \begin{pmatrix} 0.5098 \\ 0.5003 \\ -1.0004 \end{pmatrix} \qquad Q = \begin{pmatrix} 0.0711 \\ 0.5515 \\ -0.6003 \end{pmatrix}$$

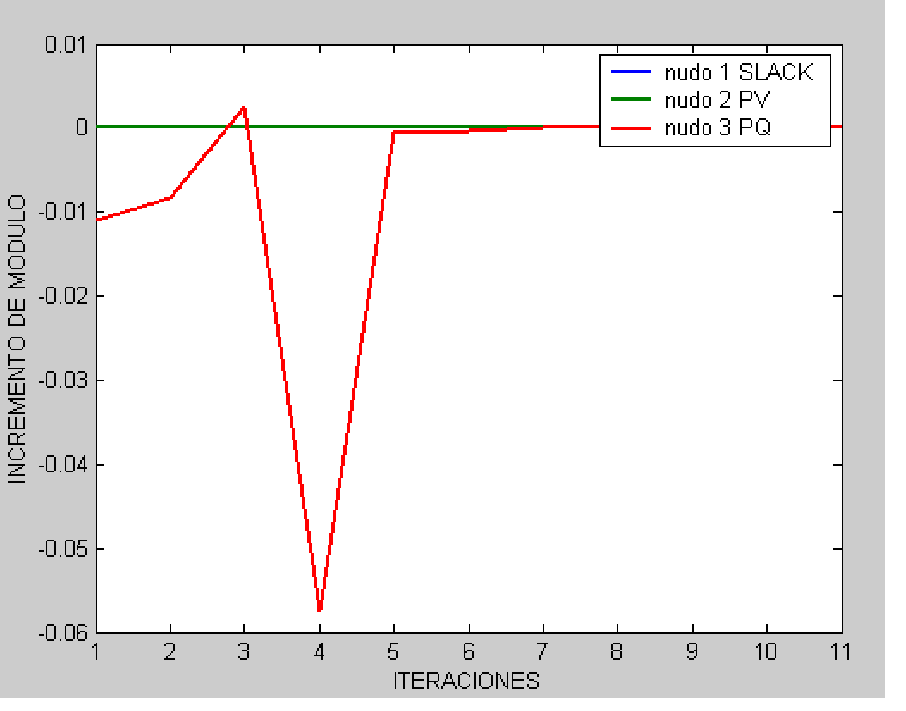
Fast decoupled load flow: Exaple
convergence





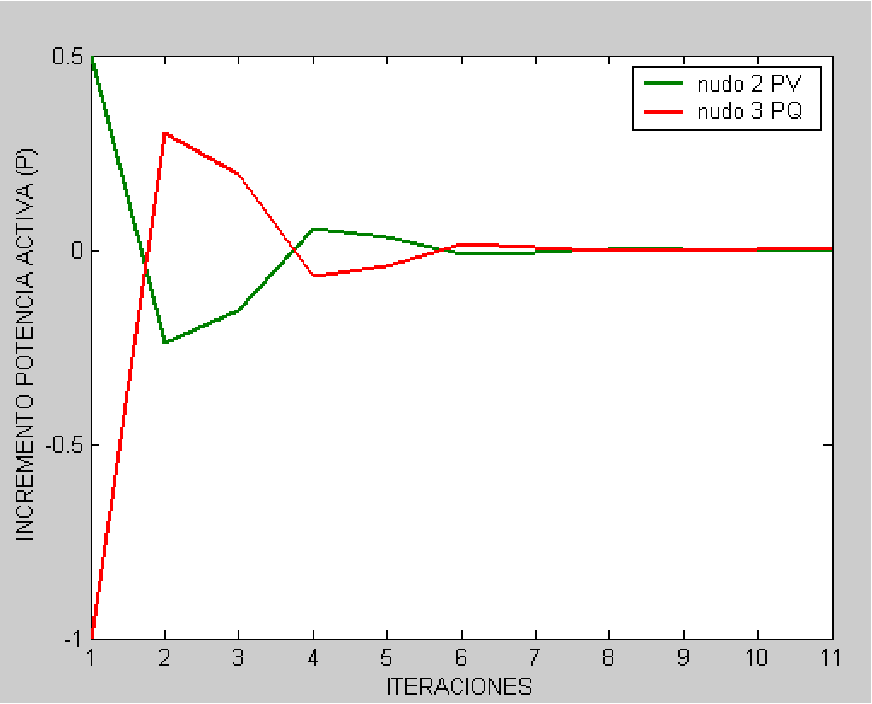
Fast decoupled load flow: Exaple convergence

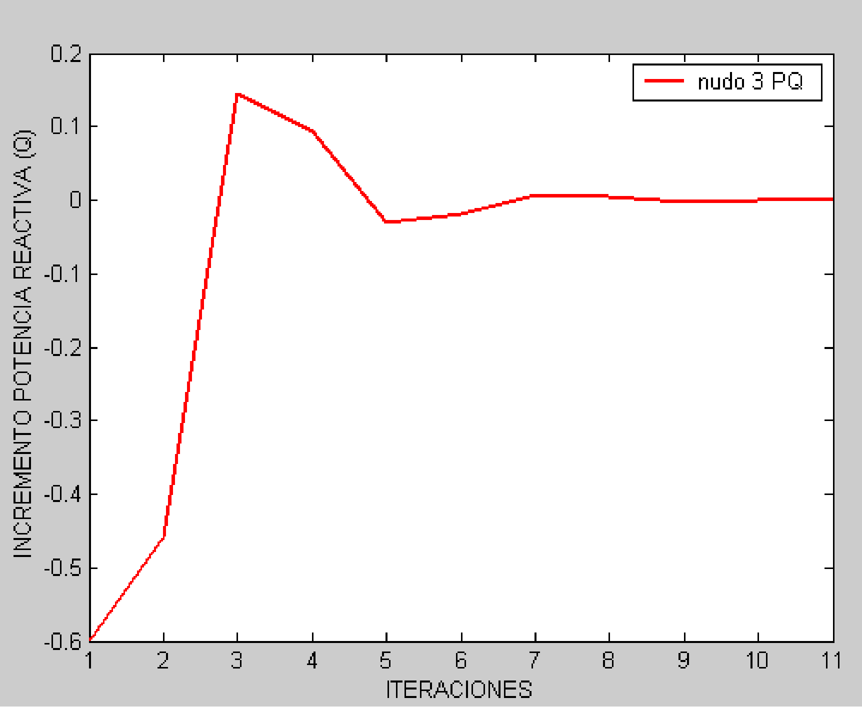




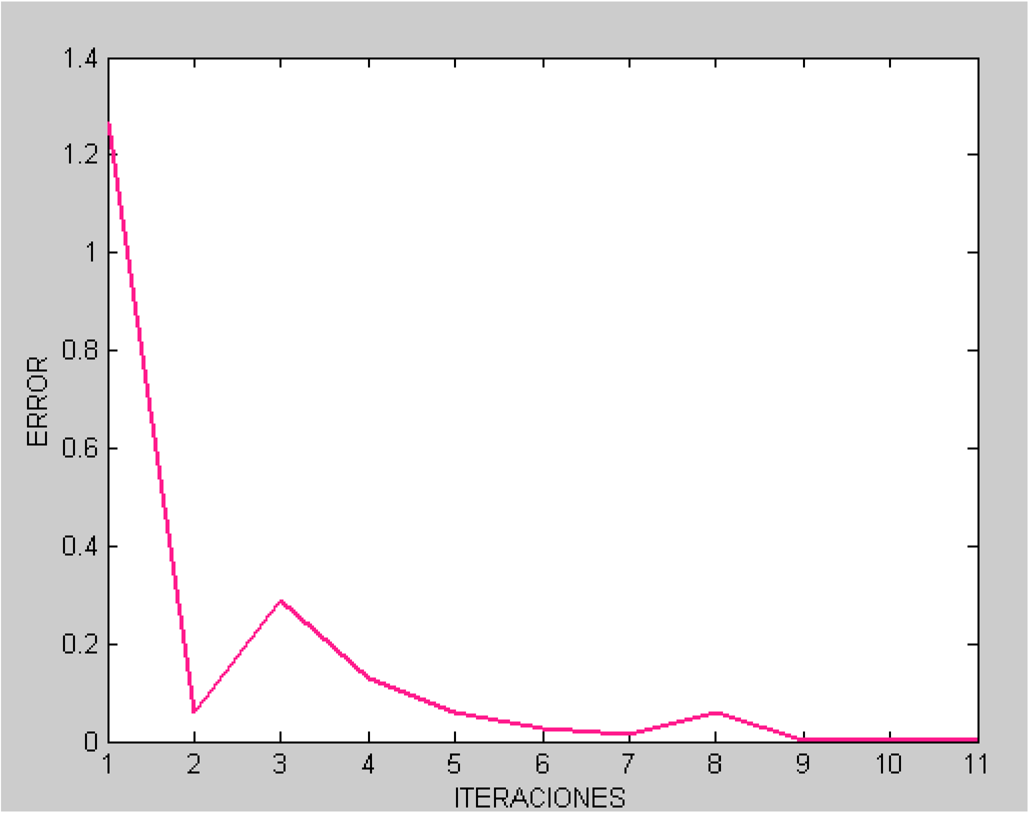
Fast decoupled load flow: Exaple

convergence





Fast decoupled load flow: Exaple convergence



6. Variable limits

Variable limits
Physical considerations

1. Voltage magnitudes out of limits
   a PQ BUS does not meet

$$V_i^{min} \leq V_i \leq V_i^{max}$$

Action: Warning! ⟶ voltage problem

2. Flow magnitudes out of limits
   A line does not meet

$$-S_{ij}^{max} \leq S_{ij} \leq S_{ij}^{max}$$

Action: Warning! ⟶ overloading of lines problem

Variable limits
Physical Considerations

3. Reactive Power out of limits

   A PV BUS does not meet

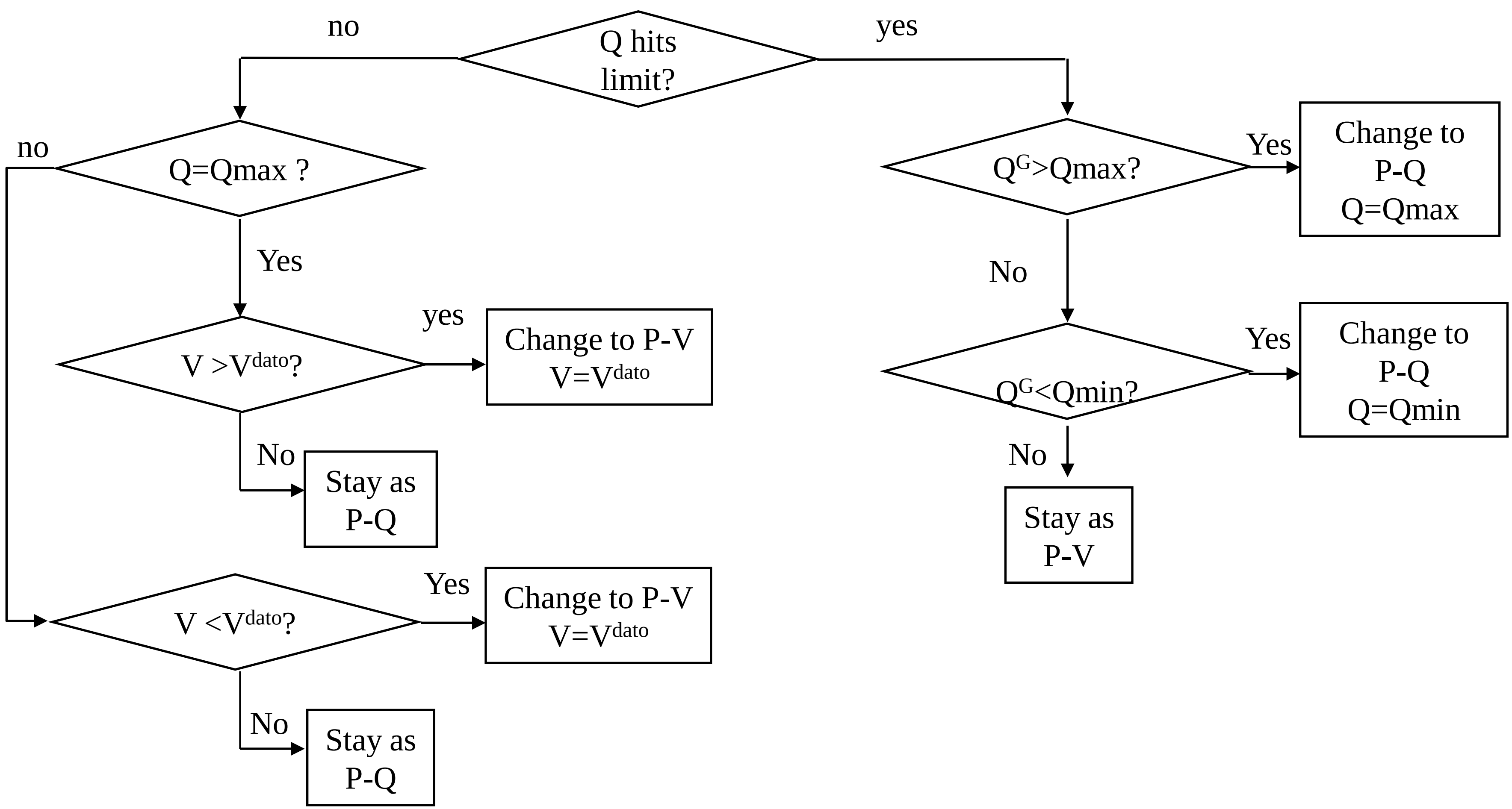
$$Q_i^{min} \leq Q_i \leq Q_i^{max}$$

Action: Wrong Formulation!

Specified voltage cannot be attained

Formulate the problem properly

Variable limits
Computational Considerations
Changing PV to PQ



7. DC load flow

DC load flow

- Approximate solution.

- Two simplifications:
  - In network model: do not consider series resistences and shunt admittances
  - Assume V_i=1 at all buses

DC load flow
Approximate analytical solution

$$P_{ij} = \frac{|\overline{V}_i||\overline{V}_j|}{X_{ij}} \times \sin\delta_{ij}$$

$$Q_{ij} = \frac{|\overline{V}_i|^2}{X_{ij}} - \frac{|\overline{V}_i||\overline{V}_j|}{X_{ij}} \times \cos\delta_{ij}$$

Assume

i) $|\overline{V}_i| \approx 1.0 \quad \forall i$

ii) $\sin\delta_{ij} \approx \delta_{ij}$

iii) $\cos\delta_{ij} \approx 1.0$

DC load flow
Approximate analytical solution

$$P_{ij} \approx \frac{\delta_{ij}}{X_{ij}} = B_{ij}\delta_{ij} = B_{ij}\delta_i - B_{ij}\delta_j$$

$$P_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij} = \left[\sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij}\right]\delta_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij}\delta_j$$

$$[P] = [B'][\delta]$$

where

$$B'_{ii} = \sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij}$$

$$B'_{ij} = -B_{ij}$$

DC load flow
Continuation

$$Q_{ij} \approx \frac{\left|\overline{V}_i\right| - \left|\overline{V}_j\right|}{X_{ij}} = B_{ij}\left(\left|\overline{V}_i\right| - \left|\overline{V}_j\right|\right) = B_{ij}\left|\overline{V}_i\right| - B_{ij}\left|\overline{V}_j\right|$$

$$Q_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} Q_{ij} = \left[\sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij}\right]\left|\overline{V}_i\right| - \sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij}\left|\overline{V}_j\right|$$

$$[Q] = \left[B^{'}\right][V]$$

DC load flow Continuation

Solution

$$[P] = [B'][\delta] \quad \text{PQ \& PV} \quad \text{Buses}$$

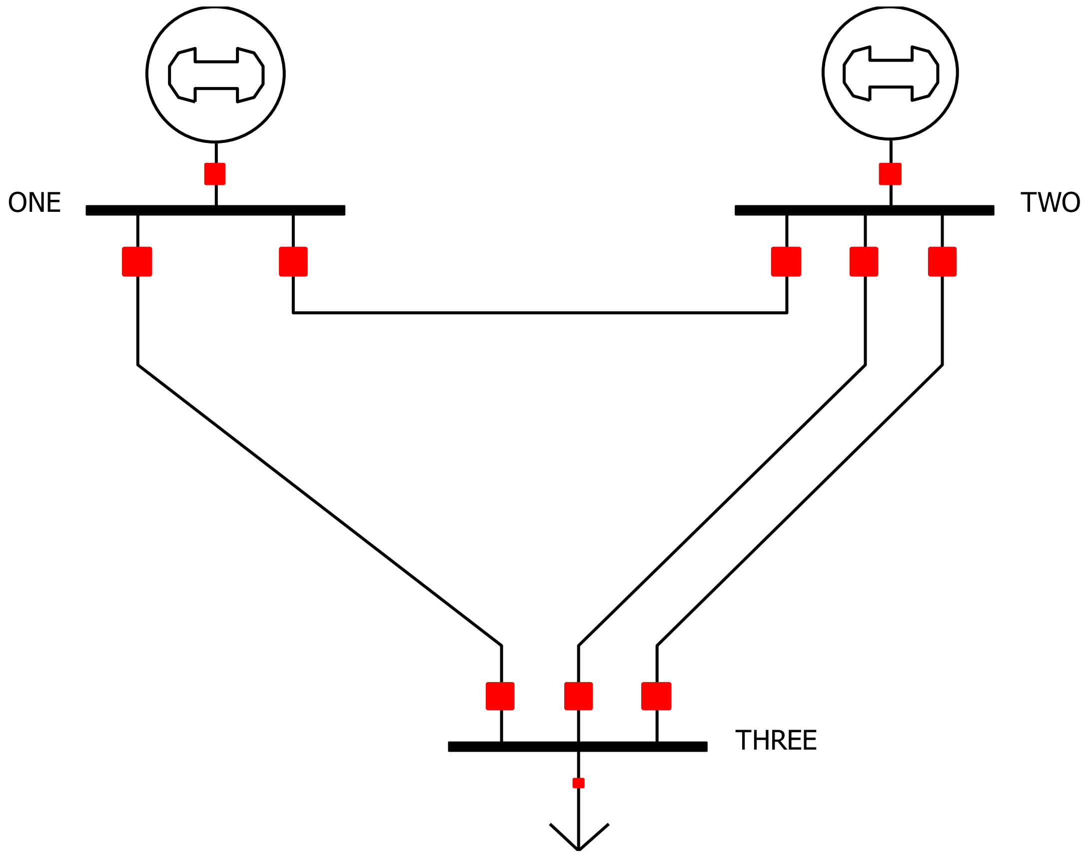
$$[Q] = [B'][V] \quad \text{PQ} \quad \text{Buses}$$

$$[\delta] = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix}; \quad [V] = \begin{bmatrix} |\overline{V}_1| \\ \vdots \\ |\overline{V}_n| \end{bmatrix}; \quad [P] = \begin{bmatrix} P_2 \\ \vdots \\ P_n \end{bmatrix}; \quad [Q] = \begin{bmatrix} Q_{m+1} \\ \vdots \\ Q_n \end{bmatrix}$$

$$[B'] = \left\{ \begin{array}{l} B'_{ii} = \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} \\ \\ B'_{ij} = -B_{ij} \end{array} \right\} \quad ; \quad B_{ij} = \frac{1}{X_{ij}}$$

DC Power flow: example

DC Power flow: example



DC Power flow: example

Data

Bus	Voltage p.u.	Power
0	1.02	-
1	1.02	PG=50MW
2	-	PC=100MW, QC=60MVAr

Line	Impedance p.u.
0-1	0.02+0.04j
0-2	0.02+0.06j
1-2	0.02+0.04j (both)

DC Power flow: example

$$B = \begin{bmatrix} 41.67 & -25 & -16.67 \\ -25 & 75 & -50 \\ -16.67 & -50 & 66.67 \end{bmatrix}$$

$$\begin{bmatrix} P_0 \\ 0.5 \\ -1.0 \end{bmatrix} = \begin{bmatrix} 41.67 & -25 & -16.67 \\ -25 & 75 & -50 \\ -16.67 & -50 & 66.67 \end{bmatrix} \begin{bmatrix} 0 \\ \delta_1 \\ \delta_2 \end{bmatrix}$$

$$\begin{bmatrix} Q_0 \\ Q_1 \\ -0.6 \end{bmatrix} = \begin{bmatrix} 41.67 & -25 & -16.67 \\ -25 & 75 & -50 \\ -16.67 & -50 & 66.67 \end{bmatrix} \begin{bmatrix} 1.02 \\ 1.02 \\ V_2 \end{bmatrix}$$

DC Power flow: example

$$\begin{pmatrix} P_0 \\ 0.5 \\ -1.0 \end{pmatrix} = \begin{pmatrix} 41.67 & -25 & -16.67 \\ -25 & 75 & -50 \\ -16.67 & -50 & 66.67 \end{pmatrix} \begin{pmatrix} 0 \\ \delta_1 \\ \delta_2 \end{pmatrix}$$

DC Power flow: example

$$Po = -25\delta_1 - 16.67\delta_2$$
$$0.5 = +75\delta_1 - 50\delta_2$$
$$-1.0 = -50\delta_1 + 66.67\delta_2$$

$$Q_0 = 41.67*1.02 - 25*1.02 - 16.67V_2$$
$$Q_1 = -25*1.02 + 75*1.02 - 50V_2$$
$$-0.6 = -16.67*1.02 - 50*1.02 + 66.67V_2$$

DC Power flow: example

Solution:

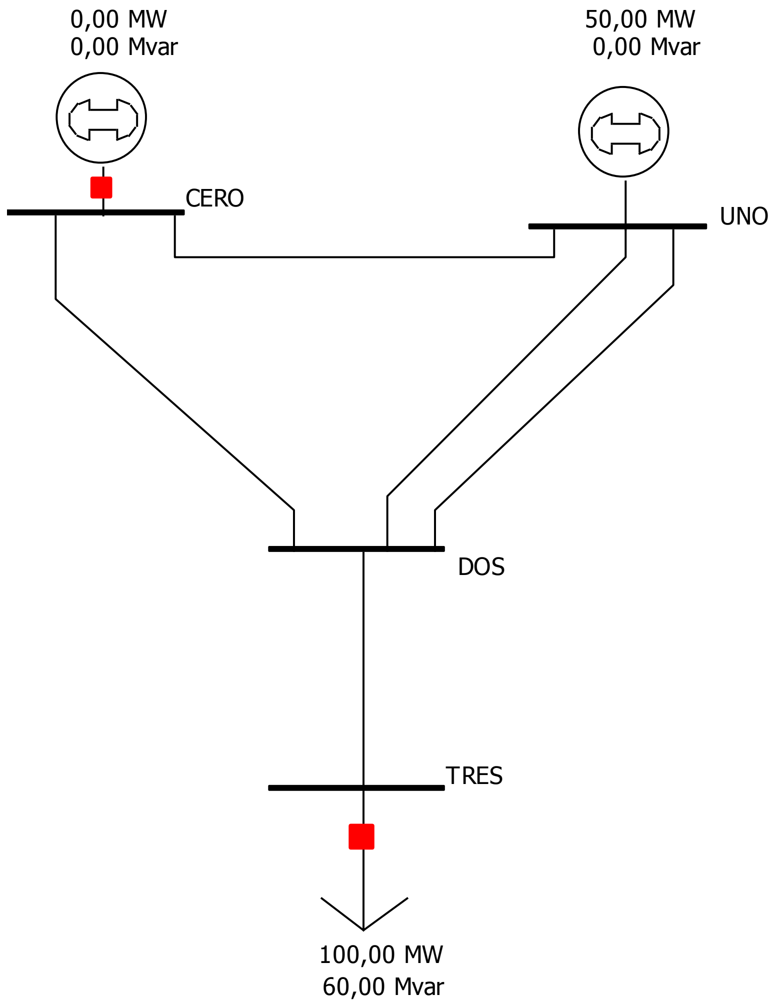
P_0	0.5p.u.=50MW
Q_0	0.15p.u.=15MVAr
Q_1	0.45p.u.=45MVAr
δ_1	-1.1459º
δ_2	-0.3819º
V_2	1.011p.u.

DC Power flow: example

PowerWorld comparison:

Var	DC	PowerWorld DC	PowerWorld G-S
P_0	0.50p.u.	0.50p.u.	0.509p.u.
Q_0	0.15p.u.	0.00p.u.	0.07p.u.
Q_1	0.45p.u.	0.00p.u.	0.55p.u.
δ_1	-0.3819	-0.3820	-0.47
δ_2	-1.1459	-1.1459	-0.96
V_2	1.0056p.u.	1.0000p.u.	1.0043p.u.

DC Power flow: example



DC Power flow: example

Data.

Bus	Voltage p.u.	Power
0	1.02	-
1	1.02	PG=50MW
2	-	PC=0MW, QC=0MVAr
3	-	PC=100MW, QC=60MVAr

Line	Impedance p.u.
0-1	0.02+0.04j
0-2	0.02+0.06j
1-2	0.02+0.04 (both)
2-3	0.1j

DC Power flow: example

$$B = \begin{bmatrix} 41.67 & -25 & -16.67 & 0 \\ -25 & 75 & -50 & 0 \\ -16.67 & -50 & 76.67 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix}$$

$$\begin{bmatrix} P_0 \\ 0.5 \\ 0.0 \\ -1.0 \end{bmatrix} = \begin{bmatrix} 41.67 & -25 & -16.67 & 0 \\ -25 & 75 & -50 & 0 \\ -16.67 & -50 & 76.67 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{bmatrix} 0 \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}$$

$$\begin{bmatrix} Q_0 \\ Q_1 \\ 0.0 \\ -0.6 \end{bmatrix} = \begin{bmatrix} 41.67 & -25 & -16.67 & 0 \\ -25 & 75 & -50 & 0 \\ -16.67 & -50 & 76.67 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{bmatrix} 1.02 \\ 1.02 \\ V_2 \\ V_3 \end{bmatrix}$$

DC Power flow: example

$$P_0 = -25\delta_1 - 16.67\delta 2 - 0^*\delta_3$$
$$0.5 = 75\delta_1 - 50\delta 2 - 0^*\delta_3$$
$$0.0 = -50\delta_1 + 76.67\delta_2 - 10\delta_3$$
$$-1.0 = -0^*\delta_1 - 10\delta_2 + 10\delta_3$$

$$Q_0 = 41.67^*1.02 - 25^*1.02 - 16.67V_2 + 0^*V_3$$
$$Q_1 = -25^*1.02 + 75^*1.02 - 50V2 + 0^*V_3$$
$$0.0 = -16.67^*1.02 - 50^*1.02 + 76.67V_2 - 10^*V_3$$
$$-0.6 = 0^*1.02 - 0^*1.02 - 10^*V_2 + 10^*V_3$$

DC Power flow: example

PowerWorld comparison:

Var	DC	PowerWorld G-S	PowerWorld DC
P_0	0.50p.u.	0.51p.u.	0.50p.u.
Q_0	0.00p.u.	0.11p.u.	0.00p.u.
Q_1	0.00p.u.	0.67p.u.	0.00p.u.
δ_1	-0.3819º	-0.48º	-0.382º
δ_2	-1.1459º	-0.91º	-1.1459º
δ_3	-6.8755º	-7.05º	-6.8755º
V_2	1.011p.u.	1.002p.u.	1.000p.u.
V_3	0.951p.u.	0.932p.u.	1.000p.u.

8. Comparison of load flow solution methods

Comparison of load flow methods

1. Gauss-Seidel (G-S)
   - ✓ Simple technique
   - ✓ Iteration time increases linearly with the number of buses. Lower iteration time than N-R. Seven times faster in large systems
   - ✓ Linear rate of convergence. Many iterations required for getting close to the solution
   - ✓ Number of iteration increases with the number of buses

Comparison of load flow methods (Continuation)

2. Newton-Raphson (N-R)

- ✓ Widely used
- ✓ Iteration-time increases linearly with the number of buses
- ✓ Quadratic rate of convergency. A few iterations for getting close to the solution
- ✓ Number of iterations independent of the number of buses of the system
- ✓ The Jacobian is a very sparse matrix
- ✓ Method non-sensitive to slack bus choice and the presence of series capacitors
- ✓ Sensitive to initial solution

Comparison of load flow methods (Continuation)

3. AC decoupled

- ✓ $[\tilde{B}]$ has to be computed and factorized only once
- ✓It requires more iterations than Newton-Raphson method
- ✓Iteration time is 5 times lower than Newton-Raphson´s iteration time
- ✓Useful for analyzing topology changes because $[\tilde{B}]$ can be easily modified
- ✓Used in planning and contigency analyses

Comparison of load flow methods (Continuation)

4. DC Decoupled

- ✓ Analytical, approximate and non-iterative method
- ✓ Good approximation for $[\delta]$, not that good approximation for $[V]$
- ✓ Used in reliability analyses
- ✓ Used in optimal pricing calculations
- ✓ Good for getting an initial point