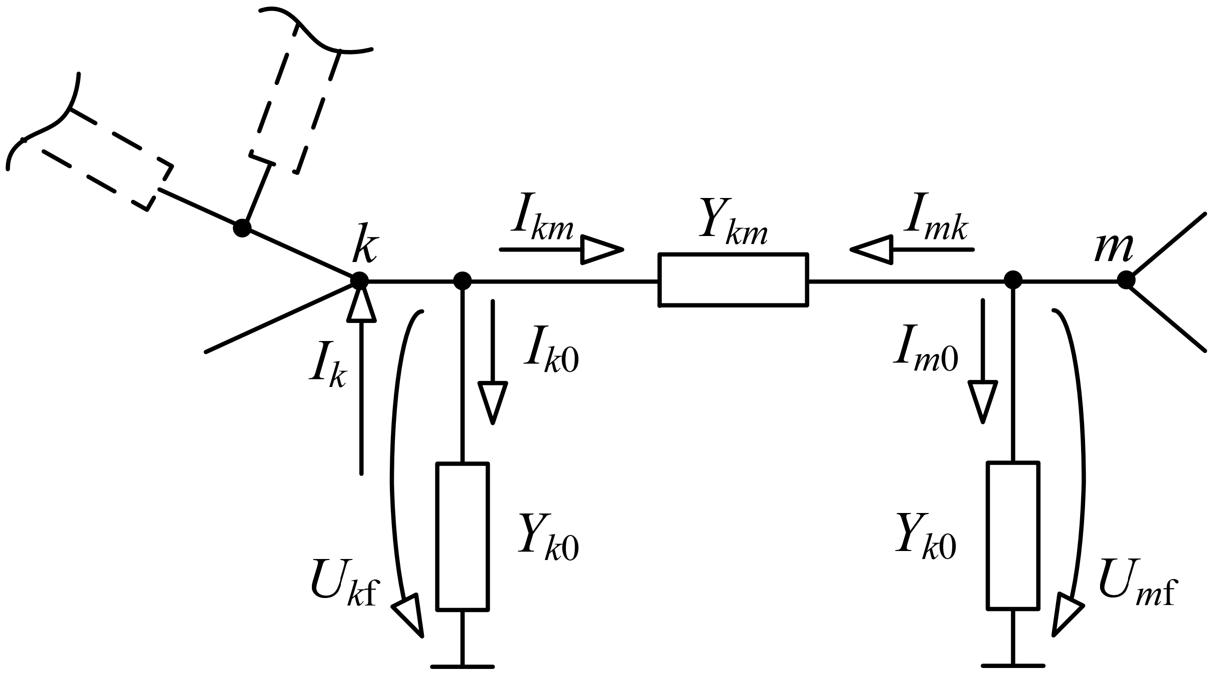
LOAD-FLOW CALCULATIONS IN MESHED SYSTEMS

Node voltage method

A system part with the node *k* and its direct neighbour *m*



$$\hat{I}_k - \sum_{\substack{m=1 \\ m \neq k}}^{n} \hat{I}_{km} - \hat{I}_{k0} = 0$$

Currents

$$\hat{I}_{km} = \left(\hat{U}_{fk} - \hat{U}_{fm}\right)\hat{Y}_{km}$$

$$\hat{I}_{k0} = \hat{U}_{fk}\hat{Y}_{k0}$$

$$\hat{I}_{k} = \sum_{\substack{m=1 \\ m\neq k}}^{n}\left(\hat{U}_{fk} - \hat{U}_{fm}\right)\hat{Y}_{km} + \hat{U}_{fk}\hat{Y}_{k0}$$

$$\hat{I}_{k} = \hat{U}_{fk}\left(\sum_{\substack{m=1 \\ m\neq k}}^{n}\hat{Y}_{km} + \hat{Y}_{k0}\right) - \sum_{\substack{m=1 \\ m\neq k}}^{n}\hat{U}_{fm}\hat{Y}_{km}$$

Let's define the node self-admittance (adm. matrix diagonal element)

$$\hat{Y}_{(k,k)} = \sum_{\substack{m=1 \\ m\neq k}}^{n}\hat{Y}_{km} + \hat{Y}_{k0} = \sum_{\substack{m=0 \\ m\neq k}}^{n}\hat{Y}_{km}$$

Node mutual admittance (non-diagonal element)

$$\hat{Y}_{(k,m)} = -\hat{Y}_{km}$$

Hence for k^{th} node current

$$\hat{I}_k = \hat{U}_{fk}\hat{Y}_{(k,k)} - \sum_{\substack{m=1 \\ m\neq k}}^{n} \hat{U}_{fm}\hat{Y}_{km} = \hat{U}_{fk}\hat{Y}_{(k,k)} + \sum_{\substack{m=1 \\ m\neq k}}^{n} \hat{U}_{fm}\hat{Y}_{(k,m)}$$

$$\hat{I}_k = \sum_{m=1}^{n} \hat{U}_{fm}\hat{Y}_{(k,m)}$$

Matrix expression

$$\left(\hat{I}\right) = \left(\hat{Y}\right)\left(\hat{U}_f\right) \qquad \sqrt{3}\left(\hat{I}\right) = \left(\hat{Y}\right)\left(\hat{U}\right)$$

Regular admittance matrix (there is at least one non-zero element $\hat{Y}_{k0}$)

$$\left(\hat{U}_f\right) = \left(\hat{Y}\right)^{-1}\left(\hat{I}\right) = \left(\hat{Z}\right)\left(\hat{I}\right)$$

Singular admittance matrix – node voltage x (1 ÷ n-1) defined

$$\begin{pmatrix} (\hat{I}_x) \\ (\hat{I}_y) \end{pmatrix} = \begin{pmatrix} (\hat{Y}_A) & (\hat{Y}_B) \\ (\hat{Y}_B)^T & (\hat{Y}_D) \end{pmatrix} \begin{pmatrix} (\hat{U}_{fx}) \\ (\hat{U}_{fy}) \end{pmatrix}$$

Hence

$$(\hat{I}_x) = (\hat{Y}_A)(\hat{U}_{fx}) + (\hat{Y}_B)(\hat{U}_{fy})$$

$$(\hat{I}_y) = (\hat{Y}_B)^T(\hat{U}_{fx}) + (\hat{Y}_D)(\hat{U}_{fy})$$

Let's calculate $(\hat{I}_x), (\hat{U}_{fy})$

$$(\hat{U}_{fy}) = (\hat{Y}_D)^{-1}(\hat{I}_y) - (\hat{Y}_D)^{-1}(\hat{Y}_B)^T(\hat{U}_{fx})$$

Gauss-Seidel method

- iterative method for non-linear equations
- not always a good convergence

Basic idea

$$f(x) \overset{!}{=} 0$$

Rewritten

$$x = g(x)$$

If $x^{(k)}$ is the estimation in kth step, the next iteration is

$$x^{(k+1)} = g(x^{(k)})$$

We continue until two following iterations difference is smaller than the prescribed precision ε

$$\left| x^{(k+1)} - x^{(k)} \right| \leq \varepsilon$$

Sometimes the convergence can be improved by the acceleration factor α (α < 1 or α > 1)

$$x^{(k+1)} = x^{(k)} + \alpha\left(g(x^{(k)}) - x^{(k)}\right)$$

The system of n equations with n unknowns

$$f_1(x_1, x_2, ..., x_n) = c_1$$

$$f_2(x_1, x_2, ..., x_n) = c_2$$

....

$$f_n(x_1, x_2, ..., x_n) = c_n$$

Each unknown is expressed from one equation

$$x_1 = c_1 + g_1(x_1, x_2, \ldots, x_n)$$

$$x_2 = c_2 + g_2(x_1, x_2, \ldots, x_n)$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$x_n = c_n + g_n(x_1, x_2, \ldots, x_n)$$

Gauss: k^{th} iteration from $(k\text{-}1)^{th}$ approximation

$$x_m^{(k)} = c_m + g_m(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_{m-1}^{(k-1)}, x_m^{(k-1)}, \ldots, x_n^{(k-1)})$$

Gauss-Seidel: for k^{th} iteration calculation also k^{th} approximations from previous equations are used

$$x_m^{(k)} = c_m + g_m(x_1^{(k)}, x_2^{(k)}, \ldots, x_{m-1}^{(k)}, x_m^{(k-1)}, \ldots, x_n^{(k-1)})$$

Convergence is tested for each variable separately.

Newton-Raphson method

- the most often method for non-linear equations
- it uses Taylor polynomial
- it converts non-linear equations solution to linear equations solution, gradually higher precision of the estimation

Basic idea

$$f(x) = c$$

If $x^{(0)}$ is the initial estimation and $\Delta x^{(0)}$ is the difference from the right solution, then

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

Taylor series

$$f(x)\Big|_{x_0} = \sum_{k=0}^{\infty} \frac{\left(\frac{df(x_0)}{dx}\right)^{(k)}}{k!} (x - x_o)^k$$

Expansion to the Taylor series

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

Higher orders neglecting (linearization)

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

where

$$\Delta c^{(0)} = c - f(x^{(0)})$$

is called “defect”.

Adding $\Delta x^{(0)}$ to the initial estimation gives the second approximation

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

(Note: impossible if the derivative equals zero)

The same relations in the next steps give the method algorithm:

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

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

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

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

The system of n equations with n unknowns

$$f_1(x_1, x_2, \ldots, x_n) = c_1$$

$$f_2(x_1, x_2, \ldots, x_n) = c_2$$

....

$$f_n(x_1, x_2, \ldots, x_n) = c_n$$

Expansion to the Taylor series

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

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

....

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

Matrix expression

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

in short

$$\left(\Delta C^{(0)}\right) = \left(J^{(0)}\right) \cdot \left(\Delta X^{(0)}\right)$$

Hence

$$\left(\Delta X^{(0)}\right) = \left(J^{(0)}\right)^{-1} \cdot \left(\Delta C^{(0)}\right)$$

The method algorithm:

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

$$\left(\Delta X^{(k)}\right) = \left(J^{(k)}\right)^{-1} \cdot \left(\Delta C^{(k)}\right)$$

$$\left(X^{(k+1)}\right) = \left(X^{(k)}\right) + \left(\Delta X^{(k)}\right)$$

$$\left(\Delta C^{(k+1)}\right)=\begin{pmatrix} c_1-(f_1^{(k+1)}) \\ c_2-(f_2^{(k+1)}) \\ \vdots \\ c_n-(f_n^{(k+1)}) \end{pmatrix}$$

where

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

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

$(J^{(k)})$ – Jakobi matrix, regularity assumption

Load Flow solution

U-I equations system can be extended to voltage-power dependence

$$\hat{I}_k = \sum_{m=1}^{n} \hat{U}_{fm} \hat{Y}_{(k,m)}$$

$$\hat{S}_k = 3\hat{S}_{fk} = 3\hat{U}_{fk}\hat{I}_k^* = 3\hat{U}_{fk} \sum_{m=1}^{n} \hat{U}_{fm}^* \hat{Y}_{(k,m)}^*$$

$$\hat{S}_k = \hat{U}_k \sum_{m=1}^{n} \hat{U}_m^* \hat{Y}_{(k,m)}^*$$

$$\left(\hat{S}\right) = \left(\hat{U}_{diag}\right)\left(\hat{Y}^*\right)\left(\hat{U}^*\right)$$

$$\begin{pmatrix} \hat{S}_1 \\ \dots \\ \hat{S}_k \\ \dots \\ \hat{S}_n \end{pmatrix} = \begin{pmatrix} \hat{U}_1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \hat{U}_k & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \hat{U}_n \end{pmatrix} \cdot \begin{pmatrix} \hat{Y}^*_{(1,1)} & \dots & \dots & \dots & \hat{Y}^*_{(1,n)} \\ \dots & \dots & \dots & \dots & \dots \\ \hat{Y}^*_{(k,1)} & \dots & \hat{Y}^*_{(k,k)} & \dots & \hat{Y}^*_{(k,n)} \\ \dots & \dots & \dots & \dots & \dots \\ \hat{Y}^*_{(n,1)} & \dots & \dots & \dots & \hat{Y}^*_{(n,n)} \end{pmatrix} \cdot \begin{pmatrix} \hat{U}^*_1 \\ \dots \\ \hat{U}^*_k \\ \dots \\ \hat{U}^*_n \end{pmatrix}$$

- powers defined → nonlinearity

Aim: to calculate **P, Q, U, δ** in nodes and branches

Note: Assumption of symmetrical system and its loading → single phase models.

Node types

Node power		Node voltage phasor components	
defined	to be calculated	defined	to be calculated
–	P, Q	U, ϑ	–
P, Q	–	–	U, ϑ
P	Q	U	ϑ
Q	P	ϑ	U

slack (swing bus) – „balance node“, balance P, Q for losses, as a huge system, large generation

PQ – loads

PU – generators, controlled voltage

Quantities

- fixed – requirements (P,Q for loads; P for generators)
- state – independent variables (U,δ for loads; δ for generators)
- control – here no changes (U for slack and generators), they change in optimization procedures

Calculations in relative values

Denominated values

$$\hat{S} = 3\hat{U}_f \hat{I}^* = \sqrt{3}\hat{U}\hat{I}^* \qquad \hat{U}_f = \hat{Z}\hat{I} \qquad \hat{U} = \sqrt{3}\hat{Z}\hat{I}$$

Base values

$$\hat{S}_v = \sqrt{3}\hat{U}_v \hat{I}_v^*$$

$$\hat{Z}_v = \frac{\hat{U}_v}{\sqrt{3}\hat{I}_v} = \frac{\hat{U}_v}{\sqrt{3}\left(\frac{\hat{S}_v}{\sqrt{3}\hat{U}_v}\right)^*} = \frac{U_v^2}{\hat{S}_v^*}$$

Relative values

$$\hat{s} \cdot S_v = \sqrt{3} \cdot \hat{u} \cdot U_v \cdot \hat{i}^* \cdot I_v^* \qquad \hat{u} \cdot U_v = \sqrt{3} \cdot \hat{z} \cdot Z_v \cdot \hat{i} \cdot I_v$$

$$\underline{\hat{s} = \hat{u} \cdot \hat{i}^*} \qquad \underline{\hat{u} = \hat{z} \cdot \hat{i}}$$

Node current (single phase)

$$\hat{I}_k = \hat{U}_{fk}\left(\sum_{\substack{m=1\\m\neq k}}^{n}\hat{Y}_{km} + \hat{Y}_{k0}\right) - \sum_{\substack{m=1\\m\neq k}}^{n}\hat{U}_{fm}\hat{Y}_{km}$$

$$\hat{i}_i = \hat{u}_i\sum_{\substack{j=0\\j\neq i}}^{n}\hat{y}_{ij} - \sum_{\substack{j=1\\j\neq i}}^{n}\hat{y}_{ij}\hat{u}_j$$

Node power

$$p_i + jq_i = \hat{u}_i \cdot \hat{i}_i^*$$

$$\hat{i}_i = \frac{p_i - jq_i}{\hat{u}_i^*}$$

hence

$$\frac{p_i - jq_i}{\hat{u}_i^*} = \hat{u}_i\sum_{\substack{j=0\\j\neq i}}^{n}\hat{y}_{ij} - \sum_{\substack{j=1\\j\neq i}}^{n}\hat{y}_{ij}\hat{u}_j$$

Gauss-Seidel Power Flow Solution

Solution for U, δ:

$$\hat{u}_i^{(k+1)} = \frac{\dfrac{p_i - jq_i}{\hat{u}_i^{*(k)}} + \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} \hat{y}_{ij}\hat{u}_j^{(k)}}{\sum\limits_{\substack{j=0 \\ j\neq i}}^{n} \hat{y}_{ij}}$$

(note: for loads P, Q < 0)

Solution for P:

$$p_i^{(k+1)} = \mathrm{Re}\left\{ \hat{u}_i^{*(k)} \left[\hat{u}_i^{(k)} \sum_{\substack{j=0 \\ j\neq i}}^{n} \hat{y}_{ij} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \hat{y}_{ij}\hat{u}_j^{(k)} \right] \right\}$$

Solution for Q:

$$q_i^{(k+1)} = -\operatorname{Im}\left\{ \hat{u}_i^{*(k)} \left[\hat{u}_i^{(k)} \sum_{\substack{j=0 \\ j\neq i}}^{n} \hat{y}_{ij} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \hat{y}_{ij}\hat{u}_j^{(k)} \right] \right\}$$

Admittance matrix elements

$$\hat{y}_{(i,i)} = \sum_{\substack{j=0 \\ j\neq i}}^{n} \hat{y}_{ij} \qquad \hat{y}_{(i,j)} = -\hat{y}_{ij}$$

PQ: U, δ slack known → 2(n-1) unknown

$$\hat{u}_i^{(k+1)} = f(p_i, q_i, \hat{u}_j^{(k)})$$

PU:

$$q_i^{(k+1)} = f(\hat{u}_i^{(k)}, \hat{u}_j^{(k)})$$

$$\hat{u}_i^{(k+1)} = f(p_i, q_i^{(k+1)}, \hat{u}_j^{(k)})$$

imaginary part taken, real part to be calculated

$$\left(e_i^{(k+1)}\right)^2 + \left(f_i^{(k+1)}\right)^2 = \left|\hat{u}_i\right|^2$$

$$e_i^{(k+1)} = \sqrt{\left|\hat{u}_i\right|^2 - \left(f_i^{(k+1)}\right)^2}$$

Newton-Raphson Power Flow Solution

$$\hat{S}_i = \hat{U}_i \sum_{j=1}^{n} \hat{U}_j^* \hat{Y}_{(i,j)}^* = U_i^2 \hat{Y}_{(i,i)}^* + \hat{U}_i \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{U}_j^* \hat{Y}_{(i,j)}^*$$

$$\hat{S}_i = f_i\left(\left(\hat{U}\right)\right)$$

Exponential form

$$\hat{S}_i = P_i + jQ_i \qquad \hat{U}_i = U_i e^{j\delta_i} \qquad \hat{Y}_{(i,j)} = Y_{(i,j)} e^{j\theta_{(i,j)}}$$

$$\hat{S}_i = U_i e^{j\delta_i} \sum_{j=1}^{n} U_j Y_{(i,j)} e^{-j\left(\delta_j + \theta_{(i,j)}\right)}$$

Power separated into the real and imaginary part

$$P_i = \sum_{j=1}^{n} U_i U_j Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$Q_i = \sum_{j=1}^{n} U_i U_j Y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

→ 2 equations for each PQ node, 1 equation for each PU node

The power changes are expressed (linearization)

$$\Delta\hat{S}_i = \sum_{j=1}^{n} \left(\frac{\partial \hat{S}_i}{\partial \delta_j} \Delta\delta_j + \frac{\partial \hat{S}_i}{\partial U_j} \Delta U_j \right)$$

$$\Delta P_i = \sum_{j=1}^{n} \left(\frac{\partial P_i}{\partial \delta_j} \Delta\delta_j + \frac{\partial P_i}{\partial U_j} \Delta U_j \right)$$

$$\Delta Q_i = \sum_{j=1}^{n} \left(\frac{\partial Q_i}{\partial \delta_j} \Delta\delta_j + \frac{\partial Q_i}{\partial U_j} \Delta U_j \right)$$

Complete equation description

$$\begin{pmatrix} \Delta P_2^{(k)} \\ \cdots \\ \Delta P_n^{(k)} \\ \Delta Q_2^{(k)} \\ \cdots \\ \Delta Q_n^{(k)} \end{pmatrix} = \begin{pmatrix} \frac{\partial P_2}{\partial \delta_2}^{(k)} & \cdots & \frac{\partial P_2}{\partial \delta_n}^{(k)} & \frac{\partial P_2}{\partial U_2}^{(k)} & \cdots & \frac{\partial P_2}{\partial U_n}^{(k)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial \delta_2}^{(k)} & \cdots & \frac{\partial P_n}{\partial \delta_n}^{(k)} & \frac{\partial P_n}{\partial U_2}^{(k)} & \cdots & \frac{\partial P_n}{\partial U_n}^{(k)} \\ \frac{\partial Q_2}{\partial \delta_2}^{(k)} & \cdots & \frac{\partial Q_2}{\partial \delta_n}^{(k)} & \frac{\partial Q_2}{\partial U_2}^{(k)} & \cdots & \frac{\partial Q_2}{\partial U_n}^{(k)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_n}{\partial \delta_2}^{(k)} & \cdots & \frac{\partial Q_n}{\partial \delta_n}^{(k)} & \frac{\partial Q_n}{\partial U_2}^{(k)} & \cdots & \frac{\partial Q_n}{\partial U_n}^{(k)} \end{pmatrix} \cdot \begin{pmatrix} \Delta \delta_2^{(k)} \\ \cdots \\ \Delta \delta_n^{(k)} \\ \Delta U_2^{(k)} \\ \cdots \\ \Delta U_n^{(k)} \end{pmatrix}$$

More compact equations form

$$\begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix} = \begin{pmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial U} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial U} \end{pmatrix} \begin{pmatrix} \Delta \delta \\ \Delta U \end{pmatrix}$$

$$(J) = \begin{pmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial U} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial U} \end{pmatrix} = \begin{pmatrix} J_1 & J_2 \\ J_3 & J_4 \end{pmatrix}$$

Equations number for *n* nodes, *s* slacks, *m* PU nodes, *p* PQ nodes (n = s + m + p):

ΔP x (n-s), ΔQ x (n-s-m)

$$P_i = \sum_{j=1}^{n} U_i U_j Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} U_i U_j Y_{(i,j)} \sin\left(-\delta_i + \delta_j + \theta_{(i,j)}\right)$$

$$\frac{\partial P_i}{\partial \delta_j} = U_i U_j Y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right) \qquad j \neq i$$

$$\frac{\partial P_i}{\partial U_i} = 2U_i Y_{(i,i)} \cos\left(\theta_{(i,i)}\right) + \sum_{\substack{j=1 \\ j\neq i}}^{n} U_j Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial P_i}{\partial U_j} = U_i Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right) \qquad j \neq i$$

$$Q_i = \sum_{j=1}^{n} U_i U_j Y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} U_i U_j Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial Q_i}{\partial \delta_j} = -U_i U_j Y_{(i,j)} \cos\left(\delta_i - \delta_j - \theta_{(i,j)}\right) \qquad j \neq i$$

$$\frac{\partial Q_i}{\partial U_i} = -2U_i Y_{(i,i)} \sin\left(\theta_{(i,i)}\right) + \sum_{\substack{j=1 \\ j\neq i}}^{n} U_j Y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial Q_i}{\partial U_j} = U_i Y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right) \qquad j \neq i$$

Iterative solution idea

$$\begin{pmatrix} \delta \\ U \end{pmatrix}_k$$

defect $\begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix}$

$$\begin{pmatrix} \Delta\delta \\ \Delta U \end{pmatrix} = (J)^{-1} \begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix}$$

$$\begin{pmatrix} \delta \\ U \end{pmatrix}_{k+1} = \begin{pmatrix} \delta \\ U \end{pmatrix}_k + \begin{pmatrix} \Delta\delta \\ \Delta U \end{pmatrix}$$

Decoupled Power Flow Solution

Transmission system: higher ration X/R for power lines
Couplings ΔP~Δδ, ΔQ~ΔU stronger than ΔP~ΔU, ΔQ~Δδ.
Therefore the Jakobi matrix can be simplified:

$$\begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix} = \begin{pmatrix} \frac{\partial P}{\partial \delta} & 0 \\ 0 & \frac{\partial Q}{\partial U} \end{pmatrix} \begin{pmatrix} \Delta \delta \\ \Delta U \end{pmatrix} = \begin{pmatrix} J_1 & 0 \\ 0 & J_4 \end{pmatrix} \begin{pmatrix} \Delta \delta \\ \Delta U \end{pmatrix}$$

So called "Decoupled problem" needs usually less time for calculation.
More iterations but quicker matrix calculations. (Number of operation for lin. equations system solution increases quicker than linearly.)
2 systems are solved sequentially in each step.
Convergence precise, only the change of Jakobi matrix, i.e. iteration steps.
Approximate solution only in case of simplified relations for P, Q.

Ideal power line (R = 0, G = 0)

$$P_{ij} = \frac{U_i U_j}{X_{ij}} \sin\delta_{ij} \qquad Q_{ij} = \frac{U_i^2}{X_{ij}} - \frac{U_i U_j}{X_{ij}} \cos\delta_{ij} - U_i^2 \cdot \frac{B}{2}$$

$$\frac{\partial P_{ij}}{\partial \delta_{ij}} = \frac{U_i U_j}{X_{ij}} \cos\delta_{ij} \qquad \frac{\partial Q_{ij}}{\partial \delta_{ij}} = \frac{U_i U_j}{X_{ij}} \sin\delta_{ij}$$

$$\frac{\partial P_{ij}}{\partial U_i} = \frac{U_j}{X_{ij}} \sin\delta_{ij} \qquad \frac{\partial Q_{ij}}{\partial U_j} = \frac{2U_i - U_j \cos\delta_{ij}}{X_{ij}}$$

For little loaded lines ($\delta_{ij} \to 0$) decoupling precise enough.

$$\frac{\partial P_{ij}}{\partial \delta_{ij}} = \frac{U_i U_j}{X_{ij}} \qquad \frac{\partial Q_{ij}}{\partial \delta_{ij}} = 0$$

$$\frac{\partial P_{ij}}{\partial U_i} = 0 \qquad \frac{\partial Q_{ij}}{\partial U_i} = \frac{2U_i - U_j}{X_{ij}}$$

The next simplifications reduce calculating $\mathbf{J_1}$ and $\mathbf{J_4}$ each iteration.

Fast Decoupled Power Flow Solution

$$\frac{\partial p_i}{\partial \delta_i} = \sum_{\substack{j=1 \\ j \neq i}}^{n} u_i u_j y_{(i,j)} \sin\left(-\delta_i + \delta_j + \theta_{(i,j)}\right)$$

$$q_i = \sum_{j=1}^{n} u_i u_j y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial p_i}{\partial \delta_i} = \sum_{j=1}^{n} u_i u_j y_{(i,j)} \sin\left(-\delta_i + \delta_j + \theta_{(i,j)}\right) - u_i^2 y_{(i,i)} \sin\left(\theta_{(i,j)}\right)$$

$$\frac{\partial p_i}{\partial \delta_i} = q_i - u_i^2 y_{(i,i)} \sin\left(\theta_{(i,j)}\right) = q_i - u_i^2 B_{(i,i)}$$

$$B_{(i,i)} = y_{(i,i)} \sin\left(\theta_{(i,j)}\right) = \mathrm{Im}\left\{y_{(i,i)}\right\}$$

Usually $B_{(i,i)} >> q_i$ and $u_i^2 \approx u_i$

$$\frac{\partial p_i}{\partial \delta_i} = -u_i B_{(i,i)}$$

$$\frac{\partial p_i}{\partial \delta_j} = u_i u_j y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

Usually $\delta_i \approx \delta_j$, $u_j \approx 1$

$$\frac{\partial p_i}{\partial \delta_j} = u_i y_{(i,j)} \sin\left(-\theta_{(i,j)}\right)$$

$$B_{(i,j)} = y_{(i,j)} \sin\left(\theta_{(i,j)}\right) = \mathrm{Im}\left\{y_{(i,j)}\right\}$$

$$\frac{\partial p_i}{\partial \delta_j} = -u_i B_{(i,j)}$$

$$\frac{\partial q_i}{\partial u_i} = -2u_i y_{(i,i)} \sin\left(\theta_{(i,i)}\right) + \sum_{\substack{j=1 \\ j\neq i}}^{n} u_j y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$q_i = \sum_{j=1}^{n} u_i u_j y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial q_i}{\partial u_i} = -u_i y_{(i,i)} \sin\left(\theta_{(i,i)}\right) + \sum_{j=1}^{n} u_j y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

$$\frac{\partial q_i}{\partial u_i} = -u_i y_{(i,i)} \sin\left(\theta_{(i,i)}\right) + q_i$$

Usually $B_{(i,i)} = y_{(i,i)} \sin\left(\theta_{(i,j)}\right) = \mathrm{Im}\left\{y_{(i,i)}\right\} >> q_i$

$$\frac{\partial q_i}{\partial u_i} = -u_i B_{(i,i)}$$

$$\frac{\partial q_i}{\partial u_j} = u_i y_{(i,j)} \sin\left(\delta_i - \delta_j - \theta_{(i,j)}\right)$$

Usually $\delta_i \approx \delta_j$

$$\frac{\partial q_i}{\partial u_j} = u_i y_{(i,j)} \sin\left(-\theta_{(i,j)}\right)$$

$$\frac{\partial q_i}{\partial u_j} = -u_i B_{(i,j)}$$

$$\begin{pmatrix} \dfrac{\Delta p}{u} \\ \dfrac{\Delta q}{u} \end{pmatrix} = -\begin{pmatrix} B' & 0 \\ 0 & B'' \end{pmatrix} \begin{pmatrix} \Delta\delta \\ \Delta u \end{pmatrix} \qquad \begin{pmatrix} \Delta\delta \\ \Delta u \end{pmatrix} = -\begin{pmatrix} B'^{-1} & 0 \\ 0 & B''^{-1} \end{pmatrix} \begin{pmatrix} \dfrac{\Delta p}{u} \\ \dfrac{\Delta q}{u} \end{pmatrix}$$

B´ and **B´´** are imaginary parts of the adm. matrix (in p.u.), their inversion is calculated only once. (Note: Division by voltages element by element.)

DC Power Flow

Relative values. Assumptions:

$$u_i \approx u_j \approx 1 \qquad \sin\delta_{ij} \approx \delta_{ij} \qquad b_{ij} = -\frac{1}{x_{ij}}$$

$$P_{ij} = \frac{U_i U_j}{X_{ij}} \sin\delta_{ij}$$

$$p_{ij} \cdot S_v = \frac{u_i \cdot U_v \cdot u_j \cdot U_v}{x_{ij} \cdot Z_v} \sin\delta_{ij}$$

$$p_{ij} = \frac{u_i \cdot u_j}{x_{ij}} \sin\delta_{ij} \;\Rightarrow\; p_{ij} = \frac{\delta_{ij}}{x_{ij}} = \frac{\delta_i - \delta_j}{x_{ij}}$$

Matrix

$$p_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\delta_i - \delta_j}{x_{ij}} = \delta_i \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{1}{x_{ij}} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\delta_j}{x_{ij}}$$

$$p_i = \delta_i b'_{(i,i)} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \delta_j b'_{(i,j)}$$

$$(p) = (b')(\delta)$$

Only longitudinal reactances → **b´** singular. 1 node as a reference with $\delta = 0$ → matrix **b´´** smaller by one order.
(DC model doesn't calculate losses, thus slack not needed but an angle reference yes.)

$$(\delta) = (b'')^{-1}(p)$$

$$\left((u) = (g)^{-1}(i)\right)$$