

TALLINNA TEHNIKAÜLIKOOL

**Elektrisüsteemi arvutamise alused
reaalajasimulatsioonide raamistikus**

Harjutused

Valminud Haridus – ja Noorteameti IT Akadeemia programmi toel

MADIS LEINAKSE, GUIDO ANDREESEN, NATHALIA DE
MORAIS DIAS CAMPOS



Tallinn 2022

Sisukord

1 Sissejuhatus	5
2 Arvutusnäited	7
2.1 Lahendataav ülesanne	7
2.2 Lahendus kasutades Gauss-Seideli meetodit	8
2.3 Lahendus kasutades Newton-Raphsoni meetodit	11
2.4 Lahendus kasutades kiiret lõhestatud meetodit	20
2.5 Võimsusvoogude ja kadude arvutus	24
3 MATLAB ülesanded	25
4 MATLAB ülesannete lahendused	55

Peatükk 1

Sissejuhatus

Käesolev harjutuste kogum "Elektrisüsteemi arvutamise alused reaalajasimulatsioonide raamistikus" koostati õppetööaine EES5030 "Energiasüsteemide optimaaljuhtimine" abimaterjalina. See valmis Haridus- ja Noorteameti IT Akadeemia programmi toel.

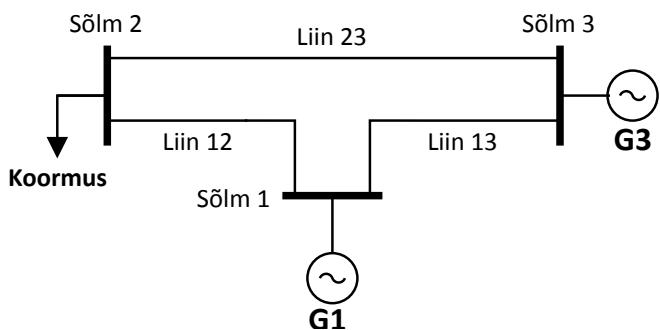
Energiasüsteemide optimaaljuhtimise raames optimeeritakse tüüpiliselt süsteemi normaal-talitlust ning arvutuste teostamisel arvutatakse püsitalitlusi. Paraku rikete ja häiringute korral võib süsteemis esineda ka raskendatud, avariiline ja avariijärgne talitlus. Nende talitluste puhul võib süsteemis esineda ka olulisi siirdeprotsesse. Elektrisüsteemide püsitalitluse ja siirdeprotsesside modelleerimiseks ning analüüsiks kasutatakse laialdaselt personaalarvuteid. Paraku on keerulisemate võrkude ning protsesside arvutamine sellisel meetodil ajamahukas. Arvutuste kiirendamiseks kasutatakse nii servereid kui ka eriotstarbelisi simulaatoreid. Osad digitaalsed simulaatorid võimaldavad süsteemi reaalajas arvutada, neid tuntakse digitaalsete reaalajasimulaatorite na. Selline arvutuskiirus saavutatakse eriotstarbelist riistvara ning elektromagnetiliste siirdeprotsesside modelleerimisalgoritmidel põhinevat tarkvara kasutades. Reaalajasimulatsioonide eripäraks on elektrisüsteemi protsesside modelleerimine sama kiirusega kui need realselt toimuks. Vastav arvutuskiirus võimaldab simulaatori abil releede ja juhtimissüsteemide testimist. Selleks ühendatakse testitavad seadmed simulaatoriga kas elektriliselt (mõõdetavad suurused pingete või vooludena) või sideliidese kaudu.

Käesolevas konspektis on elektrisüsteemi arvutamise põhimeetodite arvutusnäited. Samuti ka MATLAB harjutuste ülesanded ja lahendused. Teooriat käsitleb sama õppetööaine loengukoncept [1].

Peatükk 2

Arvutusnäited

2.1 Lahendatav ülesanne



Joonis 2.1: Lahendatav võrgumudel

Liinide näivtakistused:

$$Z_{12} = j0.025 \text{ s.ü.}$$

$$Z_{13} = j0.04 \text{ s.ü.}$$

$$Z_{23} = j0.05 \text{ s.ü.}$$

Sõlm 1 on tugisõlm, milles pinget hoitakse vääratusel $1.05\angle0^\circ$ s.ü.. Sõlm 2 on koormussõlm, millega ühendatud koormus tarbib 500 MW ja 300 Mvar. Sõlmes 3 on generaator, mis üritab hoida pinge moodulit 1.06 s.ü. ja toota 300 MW. Baasvõimsus on 100 MVA.

Tuvasta koormussõlme ping; generaatori reaktiivvõimsus ja ping nurk; tugisõlme võimsus.

2.2 Lahendus kasutades Gauss-Seideli meetodit

Arvutame juhtivused :

$$\begin{aligned}y_{12} &= \frac{1}{Z_{12}} = \frac{1}{i0.025} = -i40 \text{ s.ü.} \\y_{13} &= \frac{1}{Z_{13}} = \frac{1}{i0.04} = -i25 \text{ s.ü.} \\y_{23} &= \frac{1}{Z_{23}} = \frac{1}{i0.05} = -i20 \text{ s.ü.}\end{aligned}$$

Nende põhjal juhitlusmaatriksit koostades :

$$Y = \begin{bmatrix} y_{12} + y_{13} & -y_{12} & -y_{13} \\ -y_{12} & y_{12} + y_{23} & -y_{23} \\ -y_{13} & -y_{23} & y_{13} + y_{23} \end{bmatrix} = \begin{bmatrix} -i65 & i40 & i25 \\ i40 & -i60 & i20 \\ i25 & i20 & -i45 \end{bmatrix}$$

Normeerime võimsused (NB koormuse puhul kasutame arvutustes võimsuse negatiivset suunda!):

$$\begin{aligned}S_2 &= (-500 - i300) = -5 - i3 \text{ s.ü.} \\P_3 &= 300/100 = 3 \text{ s.ü.}\end{aligned}$$

Sisendandmetest teame, et $U_1 = 1.05\angle 0^\circ$. Teiste pingete algväärtusteks võtame (U_3 moodul lähteandmetest, teised valitud eeldatava lähedased: pingi ehk 1 s.ü. kanti võiks olla ja nurk 0 lähedane):

$$\begin{aligned}U_2^{(0)} &= 1.00\angle 0^\circ \text{ s.ü.} \\U_3^{(0)} &= 1.06\angle 0^\circ \text{ s.ü.}\end{aligned}$$

Järgnevad arvutused teostati MATLAB abil täpseid väärtsusi kasutades. Seetõttu seletustes kasutatud numbritega arvutades võivad tulemused mõningal määral erineda.

1. iteratsioon

1) Arvutame koormussõlme (PQ sõlme) pingi kasutades valemit:

$$U_k = \frac{1}{Y_{kk}} \left[\frac{P_k - jQ_k}{U_k^*} - \sum_{i=1}^m Y_{ki} U_i \right], i \neq k$$

Sõlmede koguarv on $m = 3$ ning kuna koormussõlm on 2. sõlm, siis $k = 2$.

$$U_2 = \frac{1}{Y_{22}} \left[\frac{P_2 - jQ_2}{U_2^*} - \sum_{i=1}^3 Y_{2i} U_i \right], i \neq 2$$

$$\begin{aligned}U_2^{(1)} &= \frac{1}{Y_{22}} \left[\frac{S_2^*}{U_2^*} - (Y_{21}U_1 + Y_{23}U_3) \right] = \frac{1}{-i60} \left[\frac{-5 + i3}{1.00\angle 0^\circ} - (i40 \cdot 1.05\angle 0^\circ + i20 \cdot 1.06\angle 0^\circ) \right] \\&= 1.068\angle -4.7479^\circ = 1.0033 - i0.0833 \text{ s.ü.}\end{aligned}$$

2) Arvutame generaatorisõlmes (PU sõlmes) toodetav reaktiivvoimsus kasutades valemit :

$$Q_k = -\text{Imaginääroosa} \left[U_k^* \sum_{i=1}^m Y_{ki} U_i \right]$$

$$Q_3 = -\text{Imaginääroosa} \left[U_3^* \sum_{i=1}^3 Y_{3i} U_i \right]$$

$$Q_3 = -\text{Imaginääroosa} [U_3^*(Y_{31} \cdot U_1 + Y_{32} \cdot U_2 + Y_{33} \cdot U_3)]$$

$$Q_3^{(1)} = -\text{Imaginääroosa} [1.06 \angle 0^\circ (i25 \cdot 1.05 \angle 0^\circ + i20 \cdot 1.068 \angle -4.7479^\circ - i45 \cdot 1.06 \angle 0^\circ)]$$

$$Q_3^{(1)} = 1.4663 \text{ s.ü.}$$

Kui ülesandes oleks generaatori reaktiivvoimsuse limiidid, siis Q_3 vastavust selle generaatori limiitidele kontrollitaks siin.

Seejärel arvutatakse generaatori pinge nurk δ_3 (sisuliselt pinge $U_3^{(1)}$ arvutus) kasutades valemit:

$$\delta_k^{(k+1)} = \text{Nurk} \left(\frac{1}{Y_{kk}} \left[\frac{P_k - jQ_k}{U_k^* k} - \sum_{i=1}^{k-1} Y_{ki} U_i^{(k+1)} - \sum_{i=k+1}^m Y_{ki} U_i \right] \right)$$

$$\delta_3^{(1)} = \text{Nurk} \left(\frac{1}{Y_{33}} \left[\frac{S_3^*}{U_3^*} - (Y_{31} \cdot U_1 + Y_{32} \cdot U_2) \right] \right)$$

$$\delta_3^{(1)} = \text{Nurk} \left(\frac{1}{-i45} \left[\frac{5 - i1.4663}{1.06 \angle 0^\circ} - (i25 \cdot 1.05 \angle 0^\circ + i20 \cdot 1.068 \angle -4.7479^\circ) \right] \right)$$

$$\delta_3^{(1)} = 1.3974^\circ$$

2. iteratsioon

1) Arvutame taas esmalt koormussõlme (PQ sõlme) pinge kasutades valemit:

$$U_2^{(1)} = \frac{1}{Y_{22}} \left[\frac{S_2^*}{U_2^*} - (Y_{21} U_1 + Y_{23} U_3) \right] = \frac{1}{-i60} \left[\frac{-5 + i3}{1.068 \angle 4.7479^\circ} - (i40 \cdot 1.05 \angle 0^\circ + i20 \cdot 1.06 \angle 1.3974^\circ) \right]$$

$$U_2^{(2)} = 0.9993 \angle -4.0030^\circ = 0.9969 - i0.0698$$

2) Arvutame geneeraatorisõlmes (PU sõlmes) toodetav reaktiivvoimsus kasutades valemit

$$Q_3 = -\text{Imaginääroosa} [U_3^*(Y_{31} \cdot U_1 + Y_{32} \cdot U_2 + Y_{33} \cdot U_3)]$$

$$Q_3^{(2)} = -\text{Imaginääroosa} [1.06 \angle -1.3974^\circ (i25 \cdot 1.05 \angle 0^\circ + i20 \cdot 0.9993 \angle -4.0030^\circ - i45 \cdot 1.06 \angle 1.3974^\circ)]$$

$$Q_3^{(2)} = 1.6537 \text{ s.ü.}$$

Kui ülesandes oleks generaatori reaktiivvoimsuse limiidid, siis Q_3 vastavust selle generaatori limiitidele kontrollitaks siin.

Seejärel arvutatakse generaatori pinge nurk δ_3

$$\delta_3^{(2)} = \text{Nurk} \left(\frac{1}{Y_{33}} \left[\frac{S_3^*}{U_3^*} - (Y_{31} \cdot U_1 + Y_{32} \cdot U_2) \right] \right)$$

$$\delta_3^{(2)} = \text{Nurk} \left(\frac{1}{-i45} \left[\frac{5 - i1.6537}{1.06 \angle -1.3974^\circ} - (i25 \cdot 1.05 \angle 0^\circ + i20 \cdot 0.9993 \angle -4.0030^\circ) \right] \right)$$

$$\delta_3^{(2)} = 1.7686^\circ$$

3. iteratsioon ja järgnevad

Iga iteratsioon arvutatakse esmalt PQ sõlmede pinged, seejärel kõikide generaatorisõlmede (PU sõlmede) reaktiivvõimsused ja pingete nurgad.

Peale pingete arvutuse koondumist arvutatakse tugisõlme võimsus (valemis, eeldatud, et tugisõlm on 1. sõlm):

$$\underline{S}_1 = \underline{U}_1 \underline{I}_1^* = \underline{U}_1 \sum_{i=1}^n Y_{1i} \underline{U}_i \quad (2.1)$$

Sageli arvutatakse pärast iteratsioonide koondumist võimsusvood ja kaod. Vaadeldud võrgu puhul käiks arvutus Ptk. 2.5 kohaselt.

Asjad, mida eelnevalt ei maininud

Teoorias mainisin, et ”iteratsioone teostatakse kuni lähendite muudu maksimaalne absoluutväärtus on väiksem kui valitud arvutustäpsus ϵ ehk $\max|\underline{U}^{(k+1)} - \underline{U}^{(k)}| \leq \epsilon$.“ Kui kasutada seda lõpptingimuse kontrolli meetodit, siis tuleks iga iteratsioon arvutada ka pingete muudud.

Kui kasutada lahendamisel järk-järgulist relaktsioonimeetodit, siis võrreldes Gauss-Seideli meetodiga lisandub kiirendustegur ω , mille tüüpiline väärtus on 1,3...1,7. Selle kasutamisel korrigeeritakse arvutatud pingelähendit:

$$\underline{U}_i^{(k+1)} = \underline{U}_i^{(k)} + \omega \cdot (\underline{U}_i^{(k+1),\text{valemist}} - \underline{U}_i^{(k)}) \quad (2.2)$$

2.3 Lahendus kasutades Newton-Raphsoni meetodit

Arvutame juhtivused :

$$y_{12} = \frac{1}{Z_{12}} = \frac{1}{i0.025} = -i40 \text{ s.ü.}$$

$$y_{13} = \frac{1}{Z_{13}} = \frac{1}{i0.04} = -i25 \text{ s.ü.}$$

$$y_{23} = \frac{1}{Z_{23}} = \frac{1}{i0.05} = -i20 \text{ s.ü.}$$

Nende põhjal juhitlusmaatriksit koostades :

$$Y = \begin{bmatrix} y_{12} + y_{13} & -y_{12} & -y_{13} \\ -y_{12} & y_{12} + y_{23} & -y_{23} \\ -y_{13} & -y_{23} & y_{13} + y_{23} \end{bmatrix} = \begin{bmatrix} -i65 & i40 & i25 \\ i40 & -i60 & i20 \\ i25 & i20 & -i45 \end{bmatrix}$$

Normeerime võimsused (NB koormuse puhul kasutame arvutustes võimsuse negatiivset suunda!):

$$S_2 = (-500 - i300) = -5 - i3 \text{ s.ü.}$$

$$P_3 = 300/100 = 3 \text{ s.ü.}$$

Sisendandmetest teame, et $U_1 = 1.05\angle 0^\circ$. Teiste pingete algväärtusteks võtame (U_3 moodul lähteandmetest, teised valitud eeldatava lähedased: pingi ehk 1 s.ü. kanti võiks olla ja nurk 0 lähedane):

$$U_2^{(0)} = 1.00\angle 0^\circ \text{ s.ü.}$$

$$U_3^{(0)} = 1.06\angle 0^\circ \text{ s.ü.}$$

Järgnevad arvutused teostati MATLAB abil täpseid väärtsusi kasutades. Seetõttu seletustes kasutatud numbritega arvutades võivad tulemused mõningal määral erineda.

1. iteratsioon

1) Arvutatakse koormussõlmede (**PQ** sõlmede) koormused $P_i^{(k)}$ ja $Q_i^{(k)}$ valemitega (2.7) ja (2.8).

$$P_i = |U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \cos(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.3)$$

(2.4)

$$P_2^{(1)} = |U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \cos(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$P_2^{(1)} = |U_2| \cdot [|U_1| |Y_{2,1}| \cos(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \cos(\theta_{2,2}) + |U_3| |Y_{2,3}| \cos(\theta_{2,3} - \delta_2 + \delta_3)|]$$

$$P_2^{(1)} = 1.00 \cdot \left[1.05 \cdot 40 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 60 \cos\left(\frac{-\pi}{2}\right) + 1.06 \cdot 20 \cos\left(\frac{\pi}{2} - 0 + 0\right) \right] = 0 \text{ s.ü.}$$

$$Q_i = -|U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \sin(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.5)$$

$$\begin{aligned} Q_2^{(1)} &= -|U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \sin(\theta_{2,j} - \delta_2 + \delta_j)] \\ Q_2^{(1)} &= -|U_2| \cdot [|U_1| |Y_{2,1}| \sin(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \sin(\theta_{2,2}) + |U_3| |Y_{2,3}| \sin(\theta_{2,3} - \delta_2 + \delta_3)] \\ &= -1.00 \cdot \left[1.05 \cdot 40 \sin\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 60 \sin\left(\frac{-\pi}{2}\right) + 1.06 \cdot 20 \sin\left(\frac{\pi}{2} - 0 + 0\right) \right] = -3.2 \text{ s.ü.} \end{aligned}$$

Leitakse saadud väärustuste erinevused $\Delta P_i = P_i^{oige} - P_i^{(k)}$ ja $\Delta Q_i = Q_i^{oige} - Q_i^{(k)}$ võrreldes õigete väärustega:

$$\begin{aligned} \Delta P_2 &= P_2^{oige} - P_2^{(1)} = -5 - 0 = -5 \text{ s.ü.} \\ \Delta Q_2 &= Q_2^{oige} - Q_2^{(1)} = -3 - (-3.2) = 0.2 \text{ s.ü.} \end{aligned}$$

2) Arvutatakse generaatori (PU sõlme) aktiivvõimsus $P_i^{(k)}$ valemiga (2.7) ning erinevus õigest väärustusest $\Delta P_i = P_i^{oige} - P_i^{(k)}$.

$$P_i = |U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \cos(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.6)$$

$$\begin{aligned} P_3 &= |U_3| \sum_{j=1}^m [|U_j| |Y_{3,j}| \cos(\theta_{3,j} - \delta_3 + \delta_j)] \\ P_3^{(1)} &= |U_3| \cdot [|U_1| |Y_{3,1}| \cos(\theta_{3,1} - \delta_3 + \delta_1) + |U_2| |Y_{3,2}| \cos(\theta_{3,2} - \delta_3 + \delta_2) + |U_3| |Y_{3,3}| \cos(\theta_{3,3})] \\ P_3^{(1)} &= 1.00 \cdot \left[1.05 \cdot 25 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 20 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.06 \cdot 45 \cos\left(\frac{-\pi}{2} - 0 + 0\right) \right] = 0 \text{ s.ü.} \\ \Delta P_3 &= P_3^{oige} - P_3^{(1)} = 3 - 0 = 3 \text{ s.ü.} \end{aligned}$$

3) Arvutatakse jakobiaani väärustus

Meil teadaolevad võimsused: P_2 , P_3 , Q_2 ; ning tundmatud: $|U_2|$, δ_2 , δ_3 , seega meil jakobiaan:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |U|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |U|} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \end{bmatrix}$$

Jakobiaani elementide arvutused:

J_1 diagonaali elemendid valemiga:

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{j \neq i} |U_i| |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j)$$

$$\begin{aligned}
\frac{\partial P_2}{\partial \delta_2} &= \sum_{j \neq 2} |U_2| |U_j| |Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j) \\
&= |U_2| |U_1| |Y_{21}| \sin(\theta_{21} - \delta_2 + \delta_1) + |U_2| |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) \\
&= 1.00 \cdot 1.05 \cdot 40 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) \\
&\quad = 63.2 \\
\frac{\partial P_3}{\partial \delta_3} &= \sum_{j \neq 3} |U_3| |U_j| |Y_{3j}| \sin(\theta_{3j} - \delta_3 + \delta_j) \\
&= |U_3| |U_1| |Y_{31}| \sin(\theta_{31} - \delta_3 + \delta_1) + |U_3| |U_2| |Y_{32}| \sin(\theta_{32} - \delta_3 + \delta_2) \\
&= 1.06 \cdot 1.05 \cdot 25 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) + 1.06 \cdot 1.00 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) \\
&\quad = 49.025
\end{aligned}$$

diagonaalivälised elemendid valemiga:

$$\frac{\partial P_i}{\partial \delta_j} = -|U_i| |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\begin{aligned}
\frac{\partial P_2}{\partial \delta_3} &= -|U_2| |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) = -1.00 \cdot 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) = -21.2 \\
\frac{\partial P_3}{\partial \delta_2} &= -|U_3| |U_2| |Y_{32}| \sin(\theta_{32} - \delta_3 + \delta_2) = -1.06 \cdot 1.00 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) = -21.2
\end{aligned}$$

J_2 elemendid valemiga:

$$\begin{aligned}
\frac{\partial P_i}{\partial |U_i|} &= 2|U_i| |Y_{ii}| \cos(\theta_{ii}) + \sum_{j \neq i} |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \\
\frac{\partial P_2}{\partial |U_2|} &= 2|U_2| |Y_{22}| \cos(\theta_{22}) + \sum_{j \neq 2} |U_j| |Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j) \\
&= 2|U_2| |Y_{22}| \cos(\theta_{22}) + |U_1| |Y_{21}| \cos(\theta_{21} - \delta_2 + \delta_1) + |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= 2 \cdot 1.00 \cdot 60 \cdot \cos\left(\frac{-\pi}{2}\right) + 1.05 \cdot 40 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) \\
&\quad = 0
\end{aligned}$$

$$\frac{\partial P_i}{\partial |U_j|} = |U_i| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\frac{\partial P_3}{\partial |U_2|} = |U_3| |Y_{32}| \cos(\theta_{32} - \delta_3 + \delta_2) = 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) = 0$$

J_3 elemendid valemiga:

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{j \neq i} |U_i| |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j)$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial \delta_2} &= \sum_{j \neq 2} |U_2| |U_j| |Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j) \\
&= |U_2| |U_1| |Y_{21}| \cos(\theta_{21} - \delta_2 + \delta_1) + |U_2| |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= 1.00 \cdot 1.05 \cdot 40 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) \\
&= 0
\end{aligned}$$

$$\frac{\partial Q_i}{\partial \delta_j} = -|U_i| |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial \delta_3} &= -|U_2| |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= -1.00 \cdot 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - 0 + 0\right) = 0
\end{aligned}$$

J_4 elemendid valemiga:

$$\begin{aligned}
\frac{\partial Q_i}{\partial |U_i|} &= -2|U_i| |Y_{ii}| \sin(\theta_{ii}) - \sum_{j \neq i} |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) \\
\frac{\partial Q_2}{\partial |U_2|} &= -2|U_2| |Y_{22}| \sin(\theta_{22}) - \sum_{j \neq 2} |U_j| |Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j) \\
&= -2|U_2| |Y_{22}| \sin(\theta_{22}) - |U_1| |Y_{21}| \sin(\theta_{21} - \delta_2 + \delta_1) + |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) \\
&= -2 \cdot 1.00 \cdot 60 \cdot \sin\left(-\frac{\pi}{2}\right) - 1.05 \cdot 40 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) + 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0 + 0\right) \\
&= 99.2
\end{aligned}$$

Kokkuvõttes saime jakobiaaniks

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \end{bmatrix} = \begin{bmatrix} 63.2 & -21.2 & 0 \\ -21.2 & 49.025 & 0 \\ 0 & 0 & 99.2 \end{bmatrix}$$

4) Lahendan võrrandsüsteemi:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \end{bmatrix}^{-1} \begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \\ \Delta |U_2| \end{bmatrix}$$

Näiteks jakobiaani pöördmaatriksi abil:

$$\begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \\ \Delta |U_2| \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \\ \frac{\partial \delta_2}{\partial \delta_2} & \frac{\partial \delta_3}{\partial \delta_3} & \frac{\partial |U_2|}{\partial |U_2|} \end{bmatrix}^{-1} \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta|U_2| \end{bmatrix} = \begin{bmatrix} 0.0185 & 0.0080 & 0 \\ 0.0080 & 0.0239 & 0 \\ 0 & 0 & 0.0101 \end{bmatrix} \begin{bmatrix} -5 \\ 3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} -0.0685 \\ 0.0316 \\ 0.0020 \end{bmatrix}$$

5) Arvutatakse pinge moodulid ja nurgad

Lähtudes valemitest $\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta\delta_i^{(k)}$ ja $|U_i^{(k+1)}| = |U_i^{(k)}| + \Delta|U_i^{(k)}|$

$$\delta_2^{(1)} = 0 + (-0.0685) = -0.0685 \text{ rad}$$

$$\delta_3^{(1)} = 0 + 0.0316 = 0.0316 \text{ rad}$$

$$\delta_2^{(1)} = -3.9263^\circ$$

$$\delta_3^{(1)} = 1.8082^\circ$$

$$|U_2|^{(1)} = 1.00 + 0.0020 = 1.0020 \text{ s.ü.}$$

2. iteratsioon

Korratakse samme, mida teostati esimesel iteratsioonil.

2. iteratsioonis kasutatakse eelmise iteratsiooni lõpus saadud suuruseid järgnevatel parameetritel: $\delta_2 = -0.0685$ rad, $\delta_3 = 0.0316$ rad ja $|U_2| = 1.0020$ s.ü..

1) Arvutatakse koormussõlmene (PQ sõlmede) koormused $P_i^{(k)}$ ja $Q_i^{(k)}$ valemitega (2.7) ja (2.8).

$$P_i = |U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \cos(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.7)$$

$$P_2^{(1)} = |U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \cos(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$P_2^{(1)} = |U_2| \cdot [|U_1| |Y_{2,1}| \cos(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \cos(\theta_{2,2}) + |U_3| |Y_{2,3}| \cos(\theta_{2,3} - \delta_2 + \delta_3)]$$

$$\begin{aligned} P_2^{(1)} &= 1.0020 \cdot [1.05 \cdot 40 \cos(\frac{\pi}{2} - (-0.0685) + 0) + 1.0020 \cdot 60 \cos(\frac{-\pi}{2}) + \\ &\quad 1.06 \cdot 20 \cos(\frac{\pi}{2} - (-0.0685) + 0.0316)] \\ &= -5.0043 \text{ s.ü.} \end{aligned}$$

$$Q_i = -|U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \sin(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.8)$$

$$Q_2^{(1)} = -|U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \sin(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$\begin{aligned} Q_2^{(1)} &= -|U_2| \cdot [|U_1| |Y_{2,1}| \sin(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \sin(\theta_{2,2}) + \\ &\quad |U_3| |Y_{2,3}| \sin(\theta_{2,3} - \delta_2 + \delta_3)] \end{aligned}$$

$$\begin{aligned} Q_2^{(1)} &= -1.0020 \cdot [1.05 \cdot 40 \sin(\frac{\pi}{2} - (-0.0685) + 0) + 1.0020 \cdot 60 \sin(\frac{-\pi}{2}) + \\ &\quad 1.06 \cdot 20 \sin(\frac{\pi}{2} - (-0.0685) + 0.0316)] \\ &= -2.8802 \text{ s.ü.} \end{aligned}$$

Leitakse saadud väärustuste erinevused $\Delta P_i = P_i^{oige} - P_i^{(k)}$ ja $\Delta Q_i = Q_i^{oige} - Q_i^{(k)}$ võrreldes õigete väärustega:

$$\begin{aligned}\Delta P_2 &= P_2^{oige} - P_2^{(1)} = -5 - (-5.0043) = 0.0043 \text{ s.ü.} \\ \Delta Q_2 &= Q_2^{oige} - Q_2^{(1)} = -3 - (-2.8802) = -0.1198 \text{ s.ü.}\end{aligned}$$

2) Arvutatakse generaatori (PU sõlme) aktiivvõimsus $P_i^{(k)}$ valemiga (2.7) ning erinevus õigest väärustusest $\Delta P_i = P_i^{oige} - P_i^{(k)}$.

$$P_i = |U_i| \sum_{j=1}^m [|U_j| |Y_{i,j}| \cos(\theta_{i,j} - \delta_i + \delta_j)] \quad (2.9)$$

$$\begin{aligned}P_3 &= |U_3| \sum_{j=1}^m [|U_j| |Y_{3,j}| \cos(\theta_{3,j} - \delta_3 + \delta_j)] \\ P_3^{(1)} &= |U_3| \cdot [|U_1| |Y_{3,1}| \cos(\theta_{3,1} - \delta_3 + \delta_1) + \\ &\quad |U_2| |Y_{3,2}| \cos(\theta_{3,2} - \delta_3 + \delta_2) + \\ &\quad |U_3| |Y_{3,3}| \cos(\theta_{3,3})] \\ P_3^{(1)} &= 1.00 \cdot [1.05 \cdot 25 \cos\left(\frac{\pi}{2} - 0.0316 + 0\right) + 1.0020 \cdot 20 \cos\left(\frac{\pi}{2} - 0.0316 + (-0.0685)\right) + \\ &\quad 1.06 \cdot 45 \cos\left(\frac{-\pi}{2} - 0 + 0\right)] \\ &= 3.006 \text{ s.ü.}\end{aligned}$$

$$\Delta P_3 = P_3^{oige} - P_3^{(1)} = 3 - 3.006 = -0.0006 \text{ s.ü.}$$

3) Arvutatakse jakobiaani väärus

Meil teadaolevad võimsused: P_2 , P_3 , Q_2 ; ning tundmatud: $|U_2|$, δ_2 , δ_3 , seega meil jakobiaan:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |U|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |U|} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \end{bmatrix}$$

Jakobiaani elementide arvutused:

J_1 diagonaali elemendid valemiga:

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{j \neq i} |U_i| |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j)$$

$$\begin{aligned}
\frac{\partial P_2}{\partial \delta_2} &= \sum_{j \neq 2} |U_2| |U_j| |Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j) \\
&= |U_2| |U_1| |Y_{21}| \sin(\theta_{21} - \delta_2 + \delta_1) + |U_2| |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) \\
&= 1.0020 \cdot 1.05 \cdot 40 \cdot \sin\left(\frac{\pi}{2} - (-0.0685) + 0\right) + 1.0020 \cdot 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) \\
&\quad = 63.1223 \\
\frac{\partial P_3}{\partial \delta_3} &= \sum_{j \neq 3} |U_3| |U_j| |Y_{3j}| \sin(\theta_{3j} - \delta_3 + \delta_j) \\
&= |U_3| |U_1| |Y_{31}| \sin(\theta_{31} - \delta_3 + \delta_1) + |U_3| |U_2| |Y_{32}| \sin(\theta_{32} - \delta_3 + \delta_2) \\
&= 1.06 \cdot 1.05 \cdot 25 \cdot \sin\left(\frac{\pi}{2} - 0.0316 + 0\right) + 1.06 \cdot 1.0020 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0.0316 + (-0.0685)\right) \\
&\quad = 48.9476
\end{aligned}$$

diagonaalivälised elemendid valemiga:

$$\frac{\partial P_i}{\partial \delta_j} = -|U_i| |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\begin{aligned}
\frac{\partial P_2}{\partial \delta_3} &= -|U_2| |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) = -1.0020 \cdot 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) = -21.1364 \\
\frac{\partial P_3}{\partial \delta_2} &= -|U_3| |U_2| |Y_{32}| \sin(\theta_{32} - \delta_3 + \delta_2) = -1.06 \cdot 1.0020 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - 0.0316 + (-0.0685)\right) = -21.1364
\end{aligned}$$

J_2 elemendid valemiga:

$$\begin{aligned}
\frac{\partial P_i}{\partial |U_i|} &= 2|U_i| |Y_{ii}| \cos(\theta_{ii}) + \sum_{j \neq i} |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \\
\frac{\partial P_2}{\partial |U_2|} &= 2|U_2| |Y_{22}| \cos(\theta_{22}) + \sum_{j \neq 2} |U_j| |Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j) \\
&= 2|U_2| |Y_{22}| \cos(\theta_{22}) + |U_1| |Y_{21}| \cos(\theta_{21} - \delta_2 + \delta_1) + |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= 2 \cdot 1.0020 \cdot 60 \cdot \cos(-90) + 1.05 \cdot 40 \cdot \cos\left(\frac{\pi}{2} - (-0.0685) + 0\right) + 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) \\
&\quad = -4.9942
\end{aligned}$$

$$\frac{\partial P_i}{\partial |U_j|} = |U_i| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\frac{\partial P_3}{\partial |U_2|} = |U_3| |Y_{32}| \cos(\theta_{32} - \delta_3 + \delta_2) = 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - 0.0316 + (-0.0685)\right) = 2.1183$$

J_3 elemendid valemiga:

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{j \neq i} |U_i| |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j)$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial \delta_2} &= \sum_{j \neq 2} |U_2| |U_j| |Y_{2j}| \cos(\theta_{2j} - \delta_2 + \delta_j) \\
&= |U_2| |U_1| |Y_{21}| \cos(\theta_{21} - \delta_2 + \delta_1) + |U_2| |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= 1.0020 \cdot 1.05 \cdot 40 \cdot \cos\left(\frac{\pi}{2} - (-0.0685) + 0\right) + 1.0020 \cdot 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) \\
&= -5.0043
\end{aligned}$$

$$\frac{\partial Q_i}{\partial \delta_j} = -|U_i| |U_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j), \quad j \neq i$$

$$\begin{aligned}
\frac{\partial Q_2}{\partial \delta_3} &= -|U_2| |U_3| |Y_{23}| \cos(\theta_{23} - \delta_2 + \delta_3) \\
&= -1.0020 \cdot 1.06 \cdot 20 \cdot \cos\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) = 2.1226
\end{aligned}$$

J_4 elemendid valemiga:

$$\begin{aligned}
\frac{\partial Q_i}{\partial |U_i|} &= -2|U_i| |Y_{ii}| \sin(\theta_{ii}) - \sum_{j \neq i} |U_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) \\
\frac{\partial Q_2}{\partial |U_2|} &= -2|U_2| |Y_{22}| \sin(\theta_{22}) - \sum_{j \neq 2} |U_j| |Y_{2j}| \sin(\theta_{2j} - \delta_2 + \delta_j) \\
&= -2|U_2| |Y_{22}| \sin(\theta_{22}) - |U_1| |Y_{21}| \sin(\theta_{21} - \delta_2 + \delta_1) + |U_3| |Y_{23}| \sin(\theta_{23} - \delta_2 + \delta_3) \\
&= -2 \cdot 1.0020 \cdot 60 \cdot \sin(-90) - 1.05 \cdot 40 \cdot \sin\left(\frac{\pi}{2} - (-0.0685) + 0\right) + 1.06 \cdot 20 \cdot \sin\left(\frac{\pi}{2} - (-0.0685) + 0.0316\right) \\
&= 99.4344
\end{aligned}$$

Kokkuvõttes saime jakobiaaniks

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \end{bmatrix} = \begin{bmatrix} 63.1223 & -21.1364 & -4.9942 \\ -21.1364 & 48.9476 & 2.1183 \\ -5.0043 & 2.1226 & 99.4344 \end{bmatrix}$$

4) Lahendan võrrandsüsteemi:

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \end{bmatrix}^{-1} \begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \\ \Delta |U_2| \end{bmatrix}$$

Näiteks jakobiaani pöördmaatriksi abil:

$$\begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \\ \Delta |U_2| \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |U_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |U_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |U_2|} \end{bmatrix}^{-1} \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta|U_2| \end{bmatrix} = \begin{bmatrix} 0.0186 & 0.0080 & 0.0008 \\ 0.0080 & 0.0239 & -0.001 \\ 0.0008 & -0.0001 & 0.0101 \end{bmatrix} \begin{bmatrix} 0.0043 \\ -0.0006 \\ -0.1198 \end{bmatrix} = \begin{bmatrix} -0.0000 \\ 0.0000 \\ -0.0012 \end{bmatrix}$$

5) Arvutatakse pinge moodulid ja nurgad

Lähtudes valemitest $\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta\delta_i^{(k)}$ ja $|U_i^{(k+1)}| = |U_i^{(k)}| + \Delta|U_i^{(k)}|$

$$\delta_2^{(2)} = -0.0685 + (-0.000) = -0.0685 \text{ rad}$$

$$\delta_3^{(2)} = 0.0316 + 0.0000 = 0.0316 \text{ rad}$$

$$\delta_2^{(2)} = -3.9263^\circ$$

$$\delta_3^{(2)} = 1.8083^\circ$$

$$|U_2|^{(2)} = 1.0020 - 0.0012 = 1.0008 \text{ s.ü.}$$

Järgnevad iteratsioonid

Korratakse samme, mida teostati eelmistel iteratsioonidel. Seal juures kasutatakse iga uue iteratsiooni korral eelmise iteratsiooni lõpus saadud parameetrite δ_2 , δ_3 ja $|U_2|$ suuruseid.

Iteratsioonide lõpp

Iteratsioone korratakse kuni $\Delta P_i^{(k)}$ ja $\Delta Q_i^{(k)}$ on väiksem kui määratud suurus ϵ : $\max|\Delta P_i^{(k)}| \leq \epsilon$ ja $\max|\Delta Q_i^{(k)}| \leq \epsilon$

2.4 Lahendus kasutades kiiret lõhestatud meetodit

Arvutame juhtivused

$$y_{12} = \frac{1}{Z_{12}} = \frac{1}{i0.025} = -i40 \text{ s.ü.}$$

$$y_{13} = \frac{1}{Z_{13}} = \frac{1}{i0.04} = -i25 \text{ s.ü.}$$

$$y_{23} = \frac{1}{Z_{23}} = \frac{1}{i0.05} = -i20 \text{ s.ü.}$$

Nende põhjal juhtivusmaatriks:

$$Y = \begin{bmatrix} y_{12} + y_{13} & -y_{12} & -y_{13} \\ -y_{12} & y_{12} + y_{23} & -y_{23} \\ -y_{13} & -y_{23} & y_{13} + y_{23} \end{bmatrix} = \begin{bmatrix} -i65 & i40 & i25 \\ i40 & -i60 & i20 \\ i25 & i20 & -i45 \end{bmatrix}$$

Kiire lõhestusmeetodi rakendamisel kasutatakse juhtivusmaatriksi asemel maatriksit B' ja B'' . B' saamiseks võtame Y maatriksis olevate väärustute imaginaarosad. Seejärel eemaldame tugisõlmele vastava rea ja veeru ehk esimese rea ja veeru ning saame maatriksi:

$$B' = \begin{bmatrix} -60 & 20 \\ 20 & -45 \end{bmatrix}$$

B'' saamiseks eemaldame Y maatriksis olevate väärustute imaginaarosade maatriksist tugisõlmele vastava rea ja veeru (1. rea ja 1. veeru) ning generaatorisõlmele vastava rea (3. rea) ja veeru (ka 3.). Alles jäääb ainult maatriksi keskmine element ehk $B'' = [-60]$

Normeerime võimsused (NB koormuse puhul kasutame arvutustes võimsuse negatiivset suunda!):

$$S_2 = (-500 - i300) = -5 - i3 \text{ s.ü.}$$

$$P_3 = 300/100 = 3 \text{ s.ü.}$$

Sisendandmetest teame, et $U_1 = 1.05\angle 0^\circ$. Teiste pingete algväärtusteks võtame (U_3 moodul lähteandmetest, teised valitud eeldatava lähedased: pingi ehk 1 s.ü. kanti võiks olla ja nurk 0 lähedane):

$$U_2^{(0)} = 1.00\angle 0^\circ \text{ s.ü.}$$

$$U_3^{(0)} = 1.06\angle 0^\circ \text{ s.ü.}$$

1. iteratsioon

1) Arvutatakse koormussõlmede (PQ sõlmede) koormused $P_i^{(k)}$ ja $Q_i^{(k)}$ valemitega (2.7) ja (2.8).

$$P_2^{(1)} = |U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \cos(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$P_2^{(1)} = |U_2| \cdot [|U_1| |Y_{2,1}| \cos(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \cos(\theta_{2,2}) + |U_3| |Y_{2,3}| \cos(\theta_{2,3} - \delta_2 + \delta_3)]$$

$$P_2^{(1)} = 1.00 \cdot \left[1.05 \cdot 40 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 60 \cos\left(\frac{-\pi}{2}\right) + 1.06 \cdot 20 \cos\left(\frac{\pi}{2} - 0 + 0\right) \right] = 0 \text{ s.ü.}$$

$$Q_2^{(1)} = -|U_2| \sum_{j=1}^m [|U_j||Y_{2,j}| \sin(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$Q_2^{(1)} = -|U_2| \cdot [|U_1||Y_{2,1}| \sin(\theta_{2,1} - \delta_2 + \delta_1) + |U_2||Y_{2,2}| \sin(\theta_{2,2}) + |U_3||Y_{2,3}| \sin(\theta_{2,3} - \delta_2 + \delta_3)]$$

$$Q_2^{(1)} = -1.00 \cdot \left[1.05 \cdot 40 \sin\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 60 \sin\left(\frac{-\pi}{2}\right) + 1.06 \cdot 20 \sin\left(\frac{\pi}{2} - 0 + 0\right) \right] = -3.2 \text{ s.ü.}$$

Leitakse saadud väärustuste erinevused $\Delta P_i = P_i^{oige} - P_i^{(k)}$ ja $\Delta Q_i = Q_i^{oige} - Q_i^{(k)}$ võrreldes õigete väärustega:

$$\Delta P_2 = P_2^{oige} - P_2^{(1)} = -5 - 0 = -5 \text{ s.ü.}$$

$$\Delta Q_2 = Q_2^{oige} - Q_2^{(1)} = -3 - (-3.2) = 0.2 \text{ s.ü.}$$

2) Arvutatakse generaatori (PU sõlme) aktiivvõimsus $P_i^{(k)}$ valemiga (2.7) ning erinevus õigest väärustusest $\Delta P_i = P_i^{oige} - P_i^{(k)}$.

$$P_3 = |U_3| \sum_{j=1}^m [|U_j||Y_{3,j}| \cos(\theta_{3,j} - \delta_3 + \delta_j)]$$

$$P_3^{(1)} = |U_3| \cdot [|U_1||Y_{3,1}| \cos(\theta_{3,1} - \delta_3 + \delta_1) + |U_2||Y_{3,2}| \cos(\theta_{3,2} - \delta_3 + \delta_2) + |U_3||Y_{3,3}| \cos(\theta_{3,3})]$$

$$P_3^{(1)} = 1.00 \cdot \left[1.05 \cdot 25 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.00 \cdot 20 \cos\left(\frac{\pi}{2} - 0 + 0\right) + 1.06 \cdot 45 \cos\left(\frac{-\pi}{2} - 0 + 0\right) \right] = 0 \text{ s.ü.}$$

$$\Delta P_3 = P_3^{oige} - P_3^{(1)} = 3 - 0 = 3 \text{ s.ü.}$$

3) Lahendatakse võrrandsüsteemid

$$\begin{bmatrix} \frac{\Delta P_2}{|U_2|} \\ \frac{\Delta P_3}{|U_3|} \end{bmatrix} = -[B'] \begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\Delta Q_2}{|U_2|} \end{bmatrix} = -[B''][\Delta|U_2|]$$

$$\begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \end{bmatrix} = -[B']^{-1} \begin{bmatrix} \frac{\Delta P_2}{|U_2|} \\ \frac{\Delta P_3}{|U_3|} \end{bmatrix} = -\begin{bmatrix} -60 & 20 \\ 20 & -45 \end{bmatrix}^{-1} \begin{bmatrix} \frac{-5}{1.00} \\ \frac{3}{1.06} \end{bmatrix} \quad (2.10)$$

$$= -\begin{bmatrix} -0.0196 & -0.0087 \\ -0.0087 & -0.0261 \end{bmatrix} \begin{bmatrix} \frac{-5}{1.00} \\ \frac{3}{1.06} \end{bmatrix} = \begin{bmatrix} -0.0732 \\ 0.0304 \end{bmatrix} \quad (2.11)$$

$$[\Delta|U_2|] = -[B'']^{-1} \begin{bmatrix} \Delta Q_2 \\ |U_2| \end{bmatrix} = -[-60]^{-1} \begin{bmatrix} 0.2 \\ 1.00 \end{bmatrix} = 0.0033 \quad (2.12)$$

4) Arvutatakse pinge moodulid ja nurgad

Lähtudes valemitest $\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta\delta_i^{(k)}$ ja $|U_i^{(k+1)}| = |U_i^{(k)}| + \Delta|U_i^{(k)}|$

$$\delta_2^{(1)} = 0 + (-0.0732) = -0.0732 \text{ rad}$$

$$\delta_3^{(1)} = 0 + 0.0304 = 0.0304 \text{ rad}$$

$$\delta_2^{(1)} = -4.1950^\circ$$

$$\delta_3^{(1)} = 1.7391^\circ$$

$$|U_2|^{(1)} = 1.00 + 0.0033 = 1.0033 \text{ s.ü.}$$

2. iteratsioon

Kasutatakse sama lähenemist nagu 1. iteratsioonis. Sarnaselt Newton-Raphsoni meetodile võetakse 2. iteratsioonis eelmise iteratsiooni lõpus saadud väärustused järgnevatele parameetritele:

$$\delta_2 = -0.0732 \text{ rad}, \delta_3 = 0.304 \text{ rad ja } |U_2| = 1.0033 \text{ s.ü.}$$

1) Arvutatakse koormussõlmede (PQ sõlmede) koormused $P_i^{(k)}$ ja $Q_i^{(k)}$ valemitega (2.7) ja (2.8).

$$P_2^{(1)} = |U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \cos(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$P_2^{(1)} = |U_2| \cdot [|U_1| |Y_{2,1}| \cos(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \cos(\theta_{2,2}) + |U_3| |Y_{2,3}| \cos(\theta_{2,3} - \delta_2 + \delta_3)]$$

$$P_2^{(1)} = 1.0033 \cdot [1.05 \cdot 40 \cos(\frac{\pi}{2} - (-0.0732) + 0) + 1.0033 \cdot 60 \cos(\frac{-\pi}{2}) + 1.06 \cdot 20 \cos(\frac{\pi}{2} - (-0.0732) + 0.304)] = -5.2816 \text{ s.ü.}$$

$$Q_2^{(1)} = -|U_2| \sum_{j=1}^m [|U_j| |Y_{2,j}| \sin(\theta_{2,j} - \delta_2 + \delta_j)]$$

$$Q_2^{(1)} = -|U_2| \cdot [|U_1| |Y_{2,1}| \sin(\theta_{2,1} - \delta_2 + \delta_1) + |U_2| |Y_{2,2}| \sin(\theta_{2,2}) + |U_3| |Y_{2,3}| \sin(\theta_{2,3} - \delta_2 + \delta_3)]$$

$$Q_2^{(1)} = -1.0033 \cdot [1.05 \cdot 40 \sin(\frac{\pi}{2} - (-0.0732) + 0) + 1.0033 \cdot 60 \sin(\frac{-\pi}{2}) + 1.06 \cdot 20 \sin(\frac{\pi}{2} - (-0.0732) + 0.304)] = -2.7831 \text{ s.ü.}$$

Leitakse saadud väärustuste erinevused $\Delta P_i = P_i^{oige} - P_i^{(k)}$ ja $\Delta Q_i = Q_i^{oige} - Q_i^{(k)}$ vörreldes õigete väärustega:

$$\Delta P_2 = P_2^{oige} - P_2^{(1)} = -5 - (-5.2816) = 0.2816 \text{ s.ü.}$$

$$\Delta Q_2 = Q_2^{oige} - Q_2^{(1)} = -3 - (-2.7831) = -0.2169 \text{ s.ü.}$$

2) Arvutatakse generaatori (PU sõlme) aktiivvõimsus $P_i^{(k)}$ valemiga (2.7) ning erinevus õigest väärustusest $\Delta P_i = P_i^{oige} - P_i^{(k)}$.

$$P_3 = |U_3| \sum_{j=1}^m [|U_j| |Y_{3,j}| \cos(\theta_{3,j} - \delta_3 + \delta_j)]$$

$$P_3^{(1)} = |U_3| \cdot [|U_1| |Y_{3,1}| \cos(\theta_{3,1} - \delta_3 + \delta_1) + |U_2| |Y_{3,2}| \cos(\theta_{3,2} - \delta_3 + \delta_2) + |U_3| |Y_{3,3}| \cos(\theta_{3,3})]$$

$$P_3^{(1)} = 1.0033 \cdot [1.05 \cdot 25 \cos(\frac{\pi}{2} - (-0.0732) + 0) + 1.0033 \cdot 20 \cos(\frac{\pi}{2} - 0 + 0) + 1.06 \cdot 45 \cos(\frac{-\pi}{2} - (-0.0732) + 0.304)] = 3.0435 \text{ s.ü.}$$

$$\Delta P_3 = P_3^{oige} - P_3^{(1)} = 3 - 0.0532 = -0.0435 \text{ s.ü.}$$

3) Lahendatakse võrrandsüsteemid

$$\begin{bmatrix} \frac{\Delta P_2}{|U_2|} \\ \frac{\Delta P_3}{|U_3|} \end{bmatrix} = -[B'] \begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\Delta Q_2}{|U_2|} \end{bmatrix} = -[B''][\Delta |U_2|]$$

$$\begin{bmatrix} \Delta \delta_2 \\ \Delta \delta_3 \end{bmatrix} = -[B']^{-1} \begin{bmatrix} \frac{\Delta P_2}{|U_2|} \\ \frac{\Delta P_3}{|U_3|} \end{bmatrix} = - \begin{bmatrix} -60 & 20 \\ 20 & -45 \end{bmatrix}^{-1} \begin{bmatrix} \frac{0.2816}{1.0033} \\ \frac{0.0435}{1.06} \end{bmatrix} \quad (2.13)$$

$$= - \begin{bmatrix} -0.0196 & -0.0087 \\ -0.0087 & -0.0261 \end{bmatrix} \begin{bmatrix} \frac{0.2816}{1.0033} \\ \frac{0.0435}{1.06} \end{bmatrix} = \begin{bmatrix} 0.0051 \\ 0.0014 \end{bmatrix} \quad (2.14)$$

$$[\Delta |U_2|] = -[B'']^{-1} \begin{bmatrix} \Delta Q_2 \\ |U_2| \end{bmatrix} = -[-60]^{-1} \begin{bmatrix} -0.2169 \\ 1.0033 \end{bmatrix} = -0.0036 \quad (2.15)$$

4) Arvutatakse pinge moodulid ja nurgad

Lähtudes valemitest $\delta_i^{(k+1)} = \delta_i^{(k)} + \Delta \delta_i^{(k)}$ ja $|U_i^{(k+1)}| = |U_i^{(k)}| + \Delta |U_i^{(k)}|$

$$\delta_2^{(2)} = -0.0732 + 0.0051 = -0.0681 \text{ rad}$$

$$\delta_3^{(2)} = 0.0304 + 0.0014 = 0.0317 \text{ rad}$$

$$\delta_2^{(2)} = -3.9008^\circ$$

$$\delta_3^{(2)} = 1.8176^\circ$$

$$|U_2|^{(2)} = 1.0033 - 0.0036 = 0.9997 \text{ s.ü.}$$

Järgnevad iteratsioonid

Korratakse samme, mida teostati eelmistel iteratsioonidel. Seal juures kasutatakse iga uue iteratsiooni korral eelmise iteratsiooni lõpus saadud parameetrite δ_2 , δ_3 ja $|U_2|$ suuruseid.

2.5 Võimsusvoogude ja kadude arvutus

Sageli arvutatakse pärast iteratsioonide koondumist võimsusvood ja kaod. Vaadeldud võrgu puhul käiks arvutus järgnevalt:

Arvutatakse voolud (esimene indeks näitab, millisest sõlmest vool väljub)

$$I_{12} = y_{12} \cdot (U_1 - U_2)$$

$$I_{13} = y_{13} \cdot (U_1 - U_3)$$

$$I_{23} = y_{23} \cdot (U_2 - U_3)$$

$$I_{21} = -I_{12}$$

$$I_{31} = -I_{13}$$

$$I_{32} = -I_{23}$$

Võimsusvood liinides (esimene indeks näitab, millisest sõlmest võimsus väljub)

$$S_{12} = U_1 \cdot I_{12}^*$$

$$S_{21} = U_2 \cdot I_{21}^*$$

$$S_{13} = U_1 \cdot I_{13}^*$$

$$S_{31} = U_3 \cdot I_{31}^*$$

$$S_{23} = U_2 \cdot I_{23}^*$$

$$S_{32} = U_3 \cdot I_{32}^*$$

Liinide kaod

$$S_{L12} = S_{12} + S_{21}$$

$$S_{L23} = S_{13} + S_{31}$$

$$S_{L32} = S_{23} + S_{32}$$

Peatükk 3

MATLAB ülesanded

Järgnevalt on esitatud MATLAB ülesannet kogum, mille lahendamiseks tuleb rakendada maat- rikssarvutusi, koostada süsteemide juhtivusmaatrikseid ning teostada püsitalitluse arvutusi MAT- LAB koodide abil. Nende ülesannete lahendused on esitatud peatükis 4.

Laboratory 2: Network matrix and calculations

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

Representing vectors and matrices in MATLAB

Vectors

Use the row and column vectors below to complete the tutorial.

$$V_{row} = [1 \ 2 \ 3 \ 4 \ 5]$$

$$V_{column} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

```
% Define the row vector by separating each column with a space or a comma (,)

% Define the column vector by separating each line with a semicolon (;)

% Transpose a vector. The row vector becomes a column vector, and vice versa

% Create a regularly spaced row vector. This notation is used in for loops to create an index

% Access the third element of the vectors
```

Matrices

Use the matrix M defined below to complete the tutorial.

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
% Define the matrix using a semicolon (;) to separate rows

% Access the element m23 of the matrix

% Access column 2 of the matrix
```

Matrix operations

Use the matrices M_1 and M_2 below to complete the tutorial.

$$M_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

Matrix multiplication

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \cdot \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 + b_1c_2 & a_1b_2 + b_1d_2 \\ c_1a_2 + d_1c_2 & c_1b_2 + d_1d_2 \end{bmatrix}$$

Array multiplication

Element-wise multiplication.

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \cdot \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 & b_1b_2 \\ c_1c_2 & d_1d_2 \end{bmatrix}$$

```
% Define both matrices  
  
% Add and subtract the matrices  
  
% Multiply M1 by M2  
  
% Multiply all elements of M1 by 10  
  
% Multiply M1 by M2 element-wise  
  
% Calculate the inverse of M1
```

Bus admittance matrix

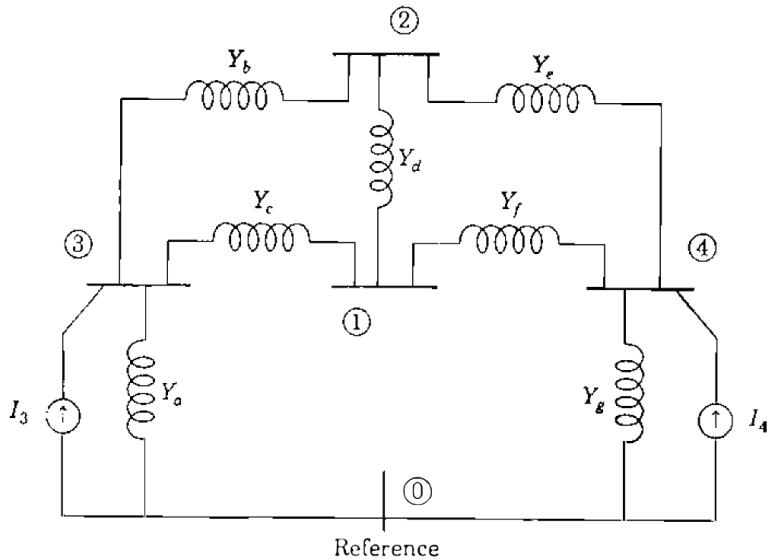
$$\mathbf{Y}_{bus} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix}$$

How to build Y_{bus}

- **Step 1:** Assign numbers to each bus. A bus is where two or more elements connect. Bus 0 should be the reference.
- **Step 2:** Create a $n \times n$ matrix, where n is the number of buses (leaving out the reference bus).
- **Step 3:** Calculate the diagonal elements Y_{ii} . Y_{ii} is the sum of the admittances connected to node i .
- **Step 4:** Calculate the off-diagonal elements Y_{ij} . Y_{ij} is the negative sum of the admittances connected between nodes i and j .

Example

Calculate the bus admittance matrix for the system below and create the matrix using MATLAB.



$$Y_{\text{bus}} =$$

$$\begin{bmatrix} & \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} \\ \textcircled{1} & (Y_c + Y_d + Y_f) & -Y_d & -Y_c & -Y_f \\ \textcircled{2} & -Y_d & (Y_b + Y_d + Y_e) & -Y_b & -Y_e \\ \textcircled{3} & -Y_c & -Y_b & (Y_a + Y_b + Y_c) & 0 \\ \textcircled{4} & -Y_f & -Y_e & 0 & (Y_e + Y_f + Y_g) \end{bmatrix}$$

Source: [1]

```
% Numeric values for the admittances
Ya = 1.25j; Yb = 0.5j; Yc = 0.2j; Yd = 1j; Ye = 0.4j; Yf = 0.8j; Yg = 0.5j;

% Define the admittance matrix Ybus
```

Impedance matrix

The impedance matrix Z_{bus} can be calculated from the bus admittance matrix as

$$Z_{\text{bus}} = Y_{\text{bus}}^{-1}$$

```
% Calculate Zbus from Ybus
```

Network solution

The network equation is written as

$$I_{\text{bus}} = Y_{\text{bus}} \cdot V_{\text{bus}}$$

This equation can be expanded resulting in

$$\begin{bmatrix} \vec{I}_1 \\ \vec{I}_2 \\ \vdots \\ \vec{I}_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \cdot \begin{bmatrix} \vec{V}_1 \\ \vec{V}_2 \\ \vdots \\ \vec{V}_n \end{bmatrix}$$

The voltages at the buses of the system can be obtained using by solving for V_{bus}

$$V_{\text{bus}} = Y_{\text{bus}}^{-1} \cdot I_{\text{bus}} \text{ or } V_{\text{bus}} = Z_{\text{bus}} \cdot I_{\text{bus}}$$

I_{bus} is computed by writing the current injections in each bus, where I_j is the current injected into bus j .

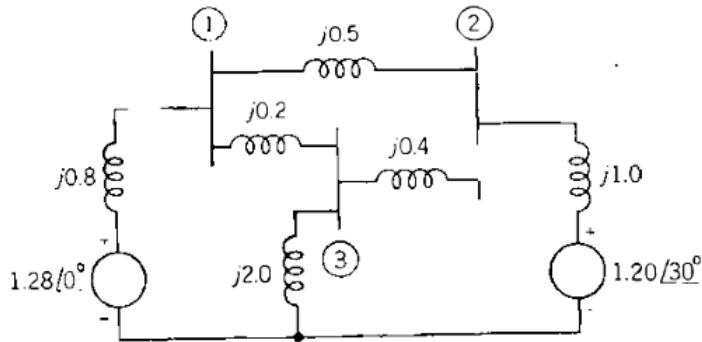
```
% Numeric values for the currents
I3 = -1.6j;
I4 = 0.6 - 1j;

% Define the current vector Ibus

% Calculate the bus voltage vector Vbus
```

Exercise

In the system below, the voltages and impedances are given in per unit. Based on this system, answer the following questions.



Source: [1]

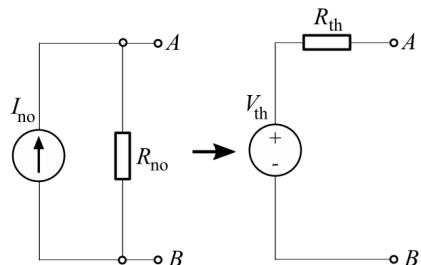
(a) Determine the bus admittance matrix Y_{bus} . Please remember the relationship between the admittance Y and the impedance Z :

$$Y = \frac{1}{Z}$$

```
% Define the diagonal elements of Ybus
% Define the off-diagonal elements of Ybus
% Define the admittance matrix Ybus using the elements calculated above
```

(b) Determine the bus impedance matrix Z_{bus} .

(c) Determine the voltage in all buses of the system. Please remember that in order to obtain the current injection vector I_{bus} you must convert the voltage sources into current sources using Norton's theorem



Source: [2]

where $I_{\text{no}} = \frac{V_{\text{th}}}{R_{\text{th}}}$ and $R_{\text{no}} = R_{\text{th}}$

```
% Define the voltage sources at buses 1 and 2
```

```
% Define the equivalent current sources at buses 1 and 2 using Norton's  
% theorem
```

```
% Define the current injection vector Ibus
```

```
% Calculate the bus voltage vector Vbus
```

(d) Calculate the complex power S supplied by each generator. Please remember that

$$S = V \cdot I^* = \frac{|V|^2}{Z^*} = |I|^2 \cdot Z$$

Laboratory 4: The Gauss-Seidel method (Cont.)

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

The power flow problem

The power flow equations are

$$P_i - jQ_i = V_i^* \sum_{k=1}^N Y_{ik} V_k$$

$$P_i = P_{gi} - P_{di}$$

$$Q_i = Q_{gi} - Q_{di}$$

Bus types are defined according to the quantities ($|V_i|$, δ_i , P_i , Q_i) specified for each bus, according to the table below.

Bus type	No. of buses	Quantities specified	No. of available equations	No. of δ_i , $ V_i $ state variables
Slack: $i = 1$	1	$\delta_1, V_1 $	0	0
Voltage controlled ($i = 2, \dots, N_g + 1$)	N_g	$P_i, V_i $	N_g	N_g
Load ($i = N_g + 2, \dots, N$)	$N - N_g - 1$	P_i, Q_i	$2(N - N_g - 1)$	$2(N - N_g - 1)$
Totals	N	$2N$	$2N - N_g - 2$	$2N - N_g - 2$

Source: [1]

Gauss-Seidel method

- **Step 1:** Define the type of each bus (load, voltage-controlled or slack bus) based on the variables known ($|V_i|$, δ_i , P_i , Q_i)
- **Step 2:** Build the admittance matrix Y_{bus}
- **Step 3:** Initialize load bus voltages (flat start \Rightarrow all voltages at $1.0\angle0^\circ$)
- **Step 4:** Define scheduled powers $P_{i,\text{sch}}$ and $Q_{i,\text{sch}}$
- **Step 5:** For each bus i ,

If i is **slack bus**, do nothing.

If i is load bus, calculate the voltage magnitude $|V_i|$ and angle δ_i using:

$$V_i^{(r)} = \frac{1}{Y_{ii}} \left\{ \frac{P_{i,sch} - jQ_{i,sch}}{V_i^{(r-1)*}} - \left[\sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} + \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

where $P_{i,sch} = P_{gi} - P_{di}$ and $Q_{i,sch} = Q_{gi} - Q_{di}$.

If using an acceleration factor α , then

$$V_{i,acc}^{(r)} = V_i^{(r-1)} + \alpha (V_i^{(r)} - V_i^{(r-1)})$$

If i is voltage-controlled bus, calculate the reactive power Q_i and voltage angle δ_i using:

$$Q_i^{(r)} = -Im \left\{ V_i^{(r-1)*} \left[\sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} + \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

$$\delta_i^{(r)} = \angle \left\{ \frac{1}{Y_{ii}} \left[\frac{P_{i,sch} - jQ_i}{V_i^{(r-1)*}} - \sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} - \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

If $Q_i^{(r)} \leq Q_{i,min}$, set $Q_i^{(r)} = Q_{i,min}$ and treat the bus as a load bus.

If $Q_i^{(r)} \geq Q_{i,max}$, set $Q_i^{(r)} = Q_{i,max}$ and treat the bus as a load bus.

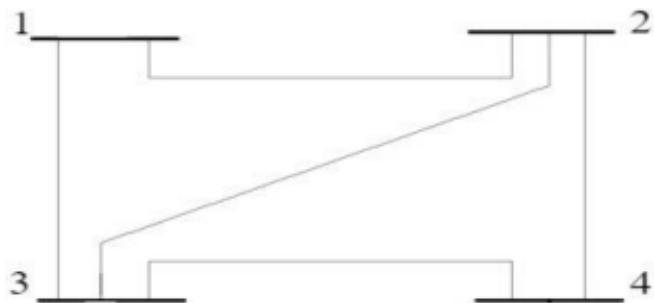
- **Step 6:** Check if algorithm converged using $|V_i^{(r)} - V_i^{(r-1)}| \leq \varepsilon$ and $|\delta_i^{(r)} - \delta_i^{(r-1)}| \leq \varepsilon$

If converged \Rightarrow calculate the power P and Q for the slack bus.

If not converged \Rightarrow repeat **Steps 5 and 6**.

Exercise 1: When PV buses are absent

Use the system below to answer the questions.



$$Y_{bus} = \begin{bmatrix} 3 - 9j & -2 + 6j & -1 + 3j & 0 \\ -2 + 6j & 3.66 - 11j & -0.66 + 2j & -1 + 3j \\ -1 + 3j & -0.66 + 2j & 3.66 - 11j & -2 + 6j \\ 0 & -1 + 3j & -2 + 6j & 3 - 9j \end{bmatrix}$$

Bus	P_{di}	Q_{di}	$ V_i $	δ_i	Bus type
1			1.04	0°	Slack
2	0.5	-0.2			PQ
3	-0.1	0.5			PQ
4	0.3	-0.1			PQ

Source: [2]

(a) Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 3; 3; 3];
```

(b) Determine the bus admittance matrix Y_{bus} .

```
Ybus = [3-9j -2+6j -1+3j 0;
       -2+6j 3.66-11j -0.66+2j -1+3j;
       -1+3j -0.66+2j 3.66-11j -2+6j;
       0 -1+3j -2+6j 3-9j];
```

(c) Define the voltage vector V_{bus} and initialize the load bus voltages.

```
Vinit = [1.04; 1; 1; 1];
```

(d) Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
P_sch = [inf; -0.5; 0.1; -0.3];

% Define the reactive power vector Q_sch
Q_sch = [inf; 0.2; -0.5; 0.1];
```

(e) Edit the file **GaussSeidel.mlx** to calculate the load bus voltages.

(f) Calculate the voltages V_{bus} in the buses using the GaussSeidel function. Use a tolerance of $\epsilon = 10^{-6}$.

```
% Define the tolerance
tolerance = 1e-6;

% Call the GaussSeidel function using the values defined previously and the
% tolerance
```

```
GaussSeidelSolution(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance, 1);
```

(g) Use the voltages vector V_{bus} below to initialize the load bus voltages. Using the same tolerance as before, what happens to the results?

```
% Initial voltage guesses  
Vinit_alt = [1.04; 4; 2; 3];  
  
% Call the GaussSeidel function
```

(h) How many iterations the method takes to converge if you use a tolerance of $\varepsilon = 10^{-3}$? Repeat the analysis for $\varepsilon = 10^{-9}$.

```
% Call the GaussSeidel function with tolerance value 1e-3  
  
% Call the GaussSeidel function with tolerance value 1e-9
```

(i) Calculate the powers P_1 and Q_1 in the slack bus. Please note that the power injected at bus 1 is given by

$$P_1 - jQ_1 = V_1^* \sum_{k=1}^N Y_{1k} V_k$$

```
% Calculate the voltages at the buses using the GaussSeidel function  
  
% Calculate P1 - jQ1  
  
% Calculate active and reactive powers P1 and Q1
```

(j) Calculate the total transmission losses. Please note that the transmission losses P_L and Q_L is given by

$$P_L = \sum_{i=1}^N P_i = \sum_{i=1}^N P_{gi} - \sum_{i=1}^N P_{di}$$

$$Q_L = \sum_{i=1}^N Q_i = \sum_{i=1}^N Q_{gi} - \sum_{i=1}^N Q_{di}$$

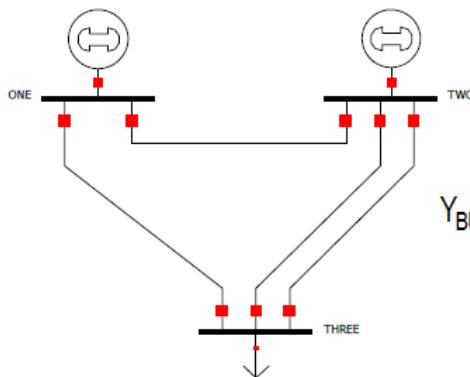
```
% Calculate the voltages at the buses using the GaussSeidel function  
  
% Calculate the total power losses PL and QL
```

(k) Edit the file **GaussSeidel.mlx** to add an acceleration factor to the calculation of the bus voltages.

(I) Run the same experiment using an acceleration factor α . What happens if you use $\alpha = 2$?

Exercise 2: When PV buses are present

Use the system below to answer the questions.



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

Bus	P_{gi}	P_{di}	Q_{di}	$ V_i $	δ_i	Bus type
1				1.02	0°	Slack
2	0.5			1.02		PV
3		1.0	0.6			PQ

Source: [3]

(a) Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

(b) Determine the bus admittance matrix Y_{bus} .

(c) Define the voltage vector V_{bus} and initialize the load bus voltages.

(d) Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
%
% Define the reactive power vector Q_sch
```

(e) Edit the file `GaussSeidel.mlx` to calculate the voltage-controlled bus reactive power and voltage angle.

(f) Calculate the voltages V_{bus} in the buses using the GaussSeidel function. Use a tolerance of $\varepsilon = 10^{-6}$.

```
% Define the tolerance  
  
% Call the GaussSeidel function using the values defined previously and the  
% tolerance
```

(g) Calculate the powers P_1 and Q_1 in the slack bus.

```
% Calculate the voltages at the buses using the GaussSeidel function  
  
% Calculate P1 - jQ1  
  
% Calculate active and reactive powers P1 and Q1
```

(h) Calculate the total transmission losses P_L and Q_L .

```
% Calculate the voltages at the buses using the GaussSeidel function  
  
% Calculate the losses PL and QL
```

```

function V = GaussSeidel(Y, V, P, Q, busTypes, tolerance, acceleration)
    % Save results for printing
    Viter(:,1) = V;

    % Total number of buses
    nbuses = size(Y,1);

    st=clock; % start the iteration time clock

    error=1;
    iteration=1;
    Vprev=V;

    % The algorithm runs until the stop condition is met
    while (error > tolerance && iteration < 500)

        % Iterate over each bus.
        for i=1:nbuses
            % If slack bus
            if busTypes(i) == 1
                continue
            end

            % Calculate the summation term over Vk * Yik
            PYV=0;
            for k=1:nbuses
                if k ~= i
                    % Calculate Vk * Yik and add it to the total
                    % YOUR ANSWER HERE
                end
            end

            % If bus is PV
            if busTypes(i) == 2
                % Estimate Qi at each iteration for the PV buses
                % YOUR ANSWER HERE
            end

            % Calculate the bus voltage
            % YOUR ANSWER HERE

            % Calculate the bus voltage considering the acceleration factor
            % YOUR ANSWER HERE

            % If bus is PV
            if busTypes(i) == 2
                % Calculate the bus voltage for the PV bus
                % YOUR ANSWER HERE
            end
        end
    end

```

```

% Tolerance at the current iteration
error = max(abs(abs(V) - abs(Vprev)));

% Save current voltage vector for next iteration
Vprev=V;

% Save results for printing
Viter(:,iteration+1)=V;

% Increment iteration count
iteration = iteration + 1;
end

ste=clock; % end the iteration time clock

% +++++++ Print results ++++++
disp('                                     Gauss Seidel Load-Flow Study')
disp('                                     Report of Power Flow Calculations ')
disp(' ')
fprintf('Number of iterations      : %g \n', iteration)
fprintf('Solution time            : %g sec.\n',etime(ste,st))
disp('Iteration    Voltages')
for i=1:size(Viter,2)
    fprintf('    %02d      ', i-1)
    for k=1:size(Viter,1)
        fprintf(' %02.3f < %02.3f      ',abs(Viter(k,i)), rad2deg(angle(Viter(k,i))))
    end
    fprintf('\n')
end
end

```

Laboratory 6: Newton-Raphson and Fast Decoupled methods

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

The power flow problem

The Newton-Raphson method calculates the bus voltages in the network using the polar form of the net active P_i and reactive Q_i powers transmitted in the grid:

$$P_i = |V_i|^2 G_{ii} + \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_i = -|V_i|^2 B_{ii} - \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

The values of the bus voltages are estimated such as to minimize the power mismatches ΔP_i and ΔQ_i :

$$\Delta P_i = P_{i,sch} - P_{i,calc}$$

$$\Delta Q_i = Q_{i,sch} - Q_{i,calc}$$

This is achieved by calculating the sensitivity of active and reactive powers in the transmission lines to changes in the voltage magnitudes and angles. This information is represented by the Jacobian and is used to make an educated decision on how to correct the voltage guesses.

$$\begin{bmatrix} \Delta \mathbf{P}_i \\ \Delta \mathbf{Q}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{P}}{\partial |\mathbf{V}|} \\ \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{Q}}{\partial |\mathbf{V}|} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\delta} \\ \Delta |\mathbf{V}| \end{bmatrix}$$

The Newton-Raphson method

Steps to implement the Newton-Raphson method:

- **Step 1:** Define the type of each bus (load, voltage-controlled or slack bus) based on the variables known ($|V_i|$, δ_i , P_i , Q_i)
- **Step 2:** Build the admittance matrix Y_{bus}
- **Step 3:** Initialize bus voltages (flat start \Rightarrow all voltages at $1.0\angle 0^\circ$)
- **Step 4:** Define scheduled powers $P_{i,sch}$ and $Q_{i,sch}$
- **Step 5:** Calculate the net transmitted powers $P_{i,calc}$ and $Q_{i,calc}$

$$P_{i,calc} = |V_i|^2 G_{ii} + \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_{i,calc} = -|V_i|^2 B_{ii} - \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

where $Y_{ik} = G_{ik} + jB_{ik}$ and $Y_{ik} = |Y_{ik}| \angle \theta_{ik}$.

- **Step 6:** Calculate the power mismatches ΔP_i and ΔQ_i

$$\Delta P_i = P_{i,sch} - P_{i,calc}$$

$$\Delta Q_i = Q_{i,sch} - Q_{i,calc}$$

- **Step 7:** Build the Jacobian matrix \mathbf{J}
- **Step 8:** Find the vector with magnitude $\Delta|V_i|$ and angle $\Delta\delta_i$ corrections:

$$\begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta|V| \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \Delta\mathbf{P}_i \\ \Delta\mathbf{Q}_i \end{bmatrix}$$

- **Step 9:** Calculate the corrected values for the voltage magnitudes and angles

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

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

- **Step 10:** Check if algorithm converged using $|V_i^{(r)} - V_i^{(r-1)}| \leq \varepsilon$ and $|\delta_i^{(r)} - \delta_i^{(r-1)}| \leq \varepsilon$

If converged \Rightarrow calculate the power P and Q for the slack bus.

If not converged \Rightarrow repeat **Step 5 through 10**.

How to build the Jacobian \mathbf{J}

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 \\ \mathbf{J}_3 & \mathbf{J}_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{P}}{\partial |V|} \\ \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{Q}}{\partial |V|} \end{bmatrix}$$

\mathbf{J}_1 elements

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial P_i}{\partial \delta_j} = -|V_i V_j Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

J₂ elements

$$\frac{\partial P_i}{\partial V_i} = 2|V_i Y_{ii}| \cos \theta_{ii} + \sum_{k=1, k \neq i}^N |V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial P_i}{\partial V_j} = |V_i Y_{ij}| \cos(\theta_{ij} + \delta_j - \delta_i)$$

J₃ elements

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial Q_i}{\partial \delta_j} = -|V_i V_j Y_{ij}| \cos(\theta_{ij} + \delta_j - \delta_i)$$

J₄ elements

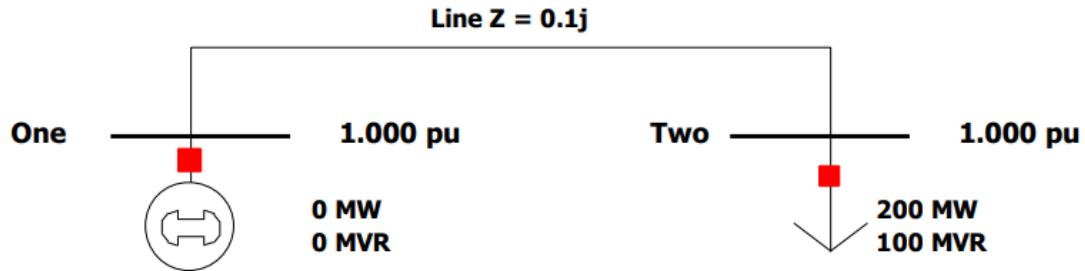
$$\frac{\partial Q_i}{\partial V_i} = -2|V_i Y_{ii}| \sin \theta_{ii} - \sum_{k=1, k \neq i}^N |V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial Q_i}{\partial V_j} = -|V_i Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

Exercise 1

Use the system below to answer the questions. Assume that $S_{\text{base}} = 100\text{MVA}$ and note that the active P_{pu} and reactive Q_{pu} power in p.u. can be obtained using

$$P_{pu} = \frac{P_{MW}}{S_{\text{base},MVA}} \text{ and } Q_{pu} = \frac{Q_{Mvar}}{S_{\text{base},MVA}}$$



- (a)** Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 3];
```

- (b)** Determine the bus admittance matrix Y_{bus} .

```
% Define the diagonal elements of Ybus
y11 = 1/0.1j;
y22 = y11;

% Define the off-diagonal elements of Ybus
y12 = -1/0.1j;
y21 = y12;

% Define the admittance matrix Ybus using the elements calculated above
Ybus = [y11 y12; y21 y22];
```

- (c)** Define the voltage vector V_{init} and initialize the bus voltages.

```
Vinit = [1; 1];
```

- (d)** Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
P_sch = [inf; -200/100];

% Define the reactive power vector Q_sch
Q_sch = [inf; -100/100];
```

- (e)** Edit the file `NewtonRaphson.mlx` to calculate the bus voltages.

- (f)** Calculate the voltages V_{bus} in the buses using the `NewtonRaphson` function. Use a tolerance of $\varepsilon = 10^{-12}$.

```
% Define the tolerance
tolerance = 1e-12;

% Call the NewtonRaphson function using the values defined previously and the
% tolerance
NewtonRaphson(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance);
```

```

deltaP =
    0      -2.0000
deltaQ =
    0      -1
J1 = 10
J2 = 1.8370e-15
J3 = 6.1232e-16
J4 = 10
Unrecognized function or variable 'corrections'.
Error in NewtonRaphson (line 227)
    corrections

```

(g) Repeat the calculation with the `GaussSeidel` function from the previous lab using the same parameters. Are there any noticeable differences?

The Fast Decoupled method

Simplifications:

- Strong relationship between $\Delta\delta \rightarrow \Delta P$ and $\Delta|V| \rightarrow \Delta Q$. This implies $J_2 = \mathbf{0}$ and $J_3 = \mathbf{0}$.
- Small angle difference between buses, $\delta_j \approx \delta_i$
- In a transmission line, $R \ll X$

The corrections vector is then written as:

$$[\Delta\delta] = J_1^{-1}[\Delta P]$$

$$[\Delta|V|] = J_4^{-1}[\Delta Q]$$

How to build the Jacobian J

$$J = \begin{bmatrix} J_1 & \mathbf{0} \\ \mathbf{0} & J_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \mathbf{0} \\ \mathbf{0} & \frac{\partial Q}{\partial |V|} \end{bmatrix}$$

J_1 elements

$$\frac{\partial P_i}{\partial \delta_i} = -|V_i|^2 B_{ii}$$

$$\frac{\partial P_i}{\partial \delta_j} = -|V_i V_j| B_{ij}$$

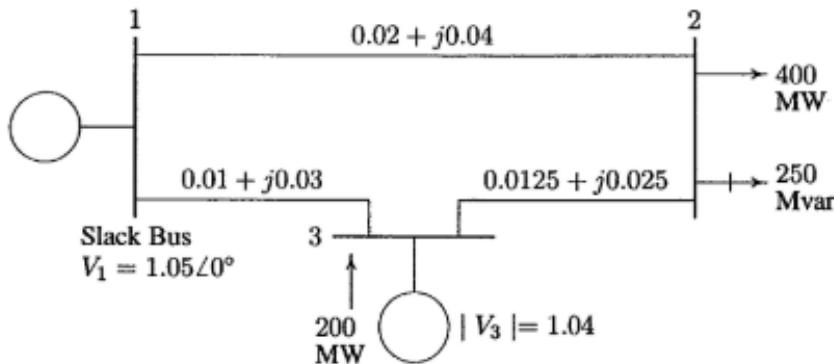
J₄ elements

$$\frac{\partial Q_i}{\partial V_i} = -|V_i| B_{ii}$$

$$\frac{\partial Q_i}{\partial V_j} = -|V_i| B_{ij}$$

Exercise 2

Use the system below to answer the questions. Assume that $S_{\text{base}} = 100\text{MVA}$ and the values provided for the transmission lines are impedances.



- (a)** Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
%
```

- (b)** Determine the bus admittance matrix Y_{bus} .

```
% Define the diagonal elements of Ybus
% Define the off-diagonal elements of Ybus
% Define the admittance matrix Ybus using the elements calculated above
```

- (c)** Define the voltage vector V_{init} and initialize the bus voltages.

```
%
```

(d) Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch  
  
% Define the reactive power vector Q_sch
```

(e) Save the **NewtonRaphson.mlx** file as **FastDecoupled.mlx** and implement the necessary changes to calculate the bus voltages using the Fast Decoupled method.

(f) Calculate the voltages V_{bus} in the buses using the **FastDecoupled** function. Use a tolerance of $\epsilon = 10^{-12}$.

(g) Compare the results with the output from the **NewtonRaphson** function.

(h) Determine the current in each transmission line. Please remember that the current between buses i and j flowing in a transmission line having an impedance Z_{line} is given by

$$I = \frac{V_i - V_j}{Z_{line}} = (V_i - V_j) \cdot Y_{line}$$

```
% Calculate the current for each line  
  
% Create a column vector Ilines with the line currents
```

(i) Determine the loading factor for each line. Are any of the lines overloaded? Please note that the loading factor is given by

$$\text{Loading} = \frac{|I|}{I_{rating}}$$

$$\text{Loading, \%} = \text{Loading} \times 100$$

```
% Current ratings for each line in p.u.  
Irating12 = 2;  
Irating13 = 2;  
Irating23 = 2;  
  
% Create a column vector Irating with the line ratings  
% Follow the same order as Ilines  
  
% Calculate the loading
```

(j) Assuming all voltages must be between 0.95 and 1.10 p.u., determine whether the system has any high or low voltage buses.

```

function V = NewtonRaphson(Y, V, P, Q, busTypes, tolerance)
DEBUG = true;

% Save results for printing
Viter(:,1) = V;

% Total number of buses
nbuses = size(Y,1);
w=0;
u=0;
for k=2:nbuses      % total no of PV busses
    if busTypes(k)==2
        w=w+1;
    end
end
for k=2:nbuses      % total no of PQ busses
    if busTypes(k)==3
        u=u+1;
    end
end

st=clock; % start the iteration time clock

error=1;
iteration=1;
Vprev=V;

% The algorithm runs until the stop condition is met
while (error > tolerance && iteration < 500)

%----- Delta P -----
for i = 2:nbuses
    VVY = 0;
    for k = 1:nbuses
        if i~=k
            % Calculate the VVY product
            % YOUR ANSWER HERE
        end
    end

    % Calculate Pcalc(i)
    % YOUR ANSWER HERE

    % Calculate the power mismatch deltaP(i)
    % YOUR ANSWER HERE

end

```

```

if (iteration == 1 && DEBUG)
    deltaP
end

%----- Delta Q -----
for i = 2:nbuses
    VVY = 0;
    for k = 1:nbuses
        if i~=k
            % Calculate the VVY product
            % YOUR ANSWER HERE
        end
    end

    % Calculate Qcalc(i)
    % YOUR ANSWER HERE

    % Calculate the power mismatch deltaQ(i)
    % YOUR ANSWER HERE
end

if (iteration == 1 && DEBUG)
    deltaQ
end

%----- P and Q mismatches -----
mismatches = [deltaP(2:end) deltaQ(2:end)];

%----- Jacobian-----
J1=zeros(nbuses-1,nbuses-1);
J2=zeros(nbuses-1,u);
J3=zeros(u,nbuses-1);
J4=zeros(u,u);

%----- J1 -----
for i=2:nbuses
    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J1(i-1,j-1)
            % YOUR ANSWER HERE
        end

        % Diagonal elements
        VVY=0;
        if i==j

```

```

        for k=1:nbuses
            if k~=i
                % Calculate the VY product
                % YOUR ANSWER HERE
            end
        end
        % Calculate the element J1(i-1,j-1)
        % YOUR ANSWER HERE
    end
end

if (iteration == 1 && DEBUG)
    J1
end

%-----J2-----%
for i=2:nbuses
    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J2(i-1,j-1)
            % YOUR ANSWER HERE
        end

        % Diagonal elements
        VY=0;
        if i==j
            for k=1:nbuses
                if k~=i
                    % Calculate the VY product
                    % YOUR ANSWER HERE
                end
            end
            % Calculate the element J2(i-1,j-1)
            % YOUR ANSWER HERE
        end
    end
end

if (iteration == 1 && DEBUG)
    J2
end

%-----J3-----%
for i=2:nbuses

```

```

for j=2:nbuses

    % Off-diagonal elements
    if i~=j
        % Calculate the element J3(i-1,j-1)
        % YOUR ANSWER HERE
    end

    % Diagonal elements
    VVY=0;
    if i==j
        for k=1:nbuses
            if k~=i
                % Calculate the VY product
                % YOUR ANSWER HERE
            end
        end
        % Calculate the element J3(i-1,j-1)
        % YOUR ANSWER HERE
    end
end
end

if (iteration == 1 && DEBUG)
    J3
end

%-----J4-----%
for i=2:nbuses

    for j=2:nbuses-w
        % Off-diagonal elements
        if i~=j
            % Calculate the element J4(i-1,j-1)
            % YOUR ANSWER HERE
        end

        % Diagonal elements
        VY=0;
        if i==j
            for k=1:nbuses
                if k~=i
                    % Calculate the VY product
                    % YOUR ANSWER HERE
                end
            end
        end
    end
end

```

```

        % Calculate the element J4(i-1,j-1)
        % YOUR ANSWER HERE
    end
end

if (iteration == 1 && DEBUG)
    J4
end

J = [J1 J2; J3 J4];

%----- Corrections vector-----

% Calculate the corrections vector corrections
% YOUR ANSWER HERE

if (iteration == 1 && DEBUG)
    corrections
end

%----- New estimates-----

for k=2:nbuses
    new_angle = angle(V(k)) + corrections(k-1);
    V(k) = abs(V(k))*exp(new_angle*1j);
end

for k=2:nbuses
    new_mag = abs(V(k)) + corrections(nbuses-2+k);
    V(k) = new_mag*exp(angle(V(k))*1j);
end

%----- Tolerance at the current iteration

error_mag = max(abs(abs(V) - abs(Vprev)));
error_angle = max(abs(angle(V) - angle(Vprev)));
error = max(error_mag, error_angle);

% Save current voltage vector for next iteration
Vprev=V;

% Save results for printing
Viter(:,iteration+1)=V;

% Increment iteration count
iteration=iteration+1;
end

```

```

ste=clock; % end the iteration time clock

% ++++++ Print results ++++++
disp('                                Newton-Raphson Load-Flow Study')
disp('                                Report of Power Flow Calculations ')
disp(' ')
fprintf('Number of iterations      : %g \n', iteration)
fprintf('Solution time            : %g sec.\n',etime(ste,st))
disp('Iteration    Voltages')
for i=1:size(Viter,2)
    fprintf('    %02d          ', i-1)
    for k=1:size(Viter,1)
        fprintf('%02.3f ∠ %02.3f    ',abs(Viter(k,i)), rad2deg(angle(Viter(k,i))))
    end
    fprintf('\n')
end

end

```

Reference: Hafiz Kashif Khaleel (2022). Newton Raphson Power Flow Solution using MATLAB (<https://www.mathworks.com/matlabcentral/fileexchange/26391-newton-raphson-power-flow-solution-using-matlab>), MATLAB Central File Exchange. Retrieved September 26, 2022.

Peatükk 4

MATLAB ülesannete lahendused

Järgnevalt on esitatud peatükis 3 esitatud MATLAB ülesannete lahendused. Esmalt maatriksarvutuse ja juhtivusmaatriksi koostamise teemaliste ülesannete. Seejärel püsitalitluse arvutused Gauss-Seideli, Newton-Raphsoni ja kiire lõhestusmeetodiga.

Laboratory 2: Network matrix and calculations

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

Representing vectors and matrices in MATLAB

Vectors

Use the row and column vectors below to complete the tutorial.

$$V_{row} = [1 \ 2 \ 3 \ 4 \ 5]$$

$$V_{column} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

```
% Define the row vector by separating each column with a space or a comma (,)  
v_row = [1 2 3 4 5];  
v_row_alt = [1, 2, 3, 4, 5];  
  
% Define the column vector by separating each line with a semicolon (;)  
v_column = [1; 2; 3; 4; 5];  
  
% Transpose a vector. The row vector becomes a column vector, and vice versa  
v_transposed = v_row.';  
v_transposed_alt = transpose(v_row);  
  
% Create a regularly spaced row vector. This notation is used in for loops to create an index  
u_row = 1:5;  
  
% Access the third element of the vectors  
v_row3 = v_row(3);  
v_column3 = v_column(3);
```

Matrices

Use the matrix M defined below to complete the tutorial.

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
% Define the matrix using a semicolon (;) to separate rows  
M = [1 2 3; 4 5 6; 7 8 9];  
  
% Access the element m23 of the matrix  
m23 = M(2,3);  
  
% Access column 2 of the matrix
```

```
m_column = M(:,2);
```

Matrix operations

Use the matrices M_1 and M_2 below to complete the tutorial.

$$M_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

Matrix multiplication

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \cdot \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 + b_1c_2 & a_1b_2 + b_1d_2 \\ c_1a_2 + d_1c_2 & c_1b_2 + d_1d_2 \end{bmatrix}$$

Array multiplication

Element-wise multiplication.

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \cdot \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 & b_1b_2 \\ c_1c_2 & d_1d_2 \end{bmatrix}$$

```
% Define both matrices
M1 = [1 2; 3 4];
M2 = [1 2; 1 2];

% Add and subtract the matrices
M_add = M1 + M2;
M_sub = M1 - M2;

% Multiply M1 by M2
M_mult = M1*M2;

% Multiply all elements of M1 by 10
M_mult_scalar = M1*10;

% Multiply M1 by M2 element-wise
M_elem = M1.*M2;

% Calculate the inverse of M1
M_inv = inv(M1);
```

Bus admittance matrix

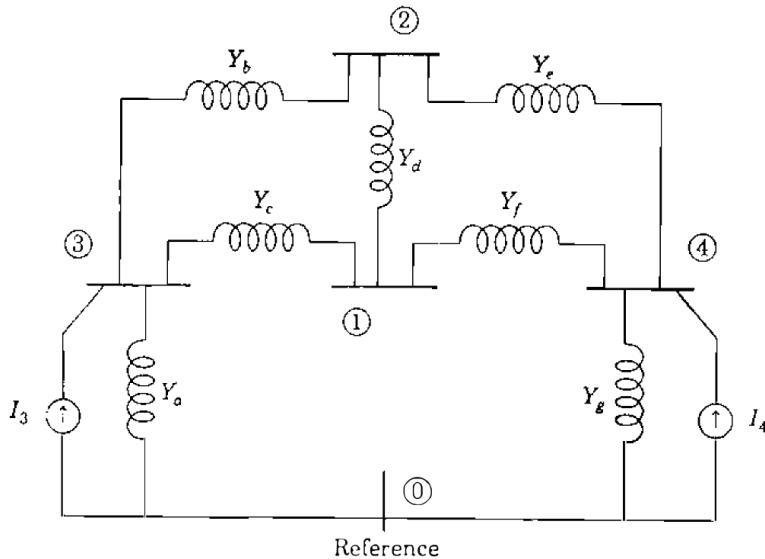
$$\mathbf{Y}_{bus} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix}$$

How to build \mathbf{Y}_{bus}

- **Step 1:** Assign numbers to each bus. A bus is where two or more elements connect. Bus 0 should be the reference.
- **Step 2:** Create a $n \times n$ matrix, where n is the number of buses (leaving out the reference bus).
- **Step 3:** Calculate the diagonal elements Y_{ii} . Y_{ii} is the sum of the admittances connected to node i .
- **Step 4:** Calculate the off-diagonal elements Y_{ij} . Y_{ij} is the negative sum of the admittances connected between nodes i and j .

Example

Calculate the bus admittance matrix for the system below and create the matrix using MATLAB.



$$\mathbf{Y}_{bus} =$$

$$\begin{bmatrix} (Y_c + Y_d + Y_f) & -Y_d & -Y_c & -Y_f \\ -Y_d & (Y_b + Y_d + Y_e) & -Y_b & -Y_e \\ -Y_c & -Y_b & (Y_a + Y_b + Y_c) & 0 \\ -Y_f & -Y_e & 0 & (Y_e + Y_f + Y_g) \end{bmatrix}$$

Source: [1]

```
% Numeric values for the admittances
Ya = 1.25j; Yb = 0.5j; Yc = 0.2j; Yd = 1j; Ye = 0.4j; Yf = 0.8j; Yg = 0.5j;

% Define the admittance matrix Ybus
Ybus = [(Yc+Yd+Yf) -Yd -Yc -Yf;
```

```

-Yd (Yb+Yd+Ye) -Yb -Ye;
-Yc -Yb (Ya+Yb+Yc) 0;
-Yf -Ye 0 (Ye+Yf+Yg)];

```

Impedance matrix

The impedance matrix Z_{bus} can be calculated from the bus admittance matrix as

$$Z_{\text{bus}} = Y_{\text{bus}}^{-1}$$

```

% Calculate Zbus from Ybus
Zbus = inv(Ybus);

```

Network solution

The network equation is written as

$$I_{\text{bus}} = Y_{\text{bus}} \cdot V_{\text{bus}}$$

This equation can be expanded resulting in

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}$$

The voltages at the buses of the system can be obtained using by solving for V_{bus}

$$V_{\text{bus}} = Y_{\text{bus}}^{-1} \cdot I_{\text{bus}} \text{ or } V_{\text{bus}} = Z_{\text{bus}} \cdot I_{\text{bus}}$$

I_{bus} is computed by writing the current injections in each bus, where I_j is the current injected into bus j .

```

% Numeric values for the currents
I3 = -1.6j;
I4 = 0.6 - 1j;

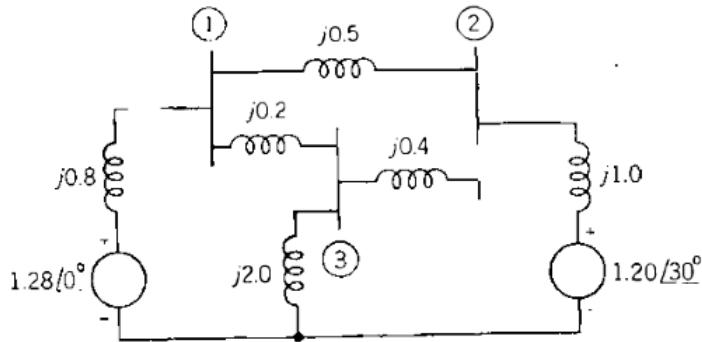
% Define the current vector Ibus
Ibus = [0; 0; I3; I4];

% Calculate the bus voltage vector Vbus
Vbus = Zbus*Ibus; % Alternatively, Vbus = Ybus\Ibus;

```

Exercise

In the system below, the voltages and **impedances** are given in per unit. Based on this system, answer the following questions.



Source: [1]

- (a)** Determine the bus admittance matrix Y_{bus} . Please remember the relationship between the admittance Y and the impedance Z :

$$Y = \frac{1}{Z}$$

```
% Define the diagonal elements of Ybus
y11 = 1/0.8j + 1/0.2j + 1/0.5j;
y22 = 1/0.5j + 1/0.4j + 1/1j;
y33 = 1/0.2j + 1/0.4j + 1/2j;

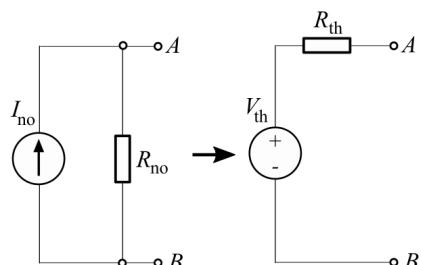
% Define the off-diagonal elements of Ybus
y12 = -1/0.5j;
y21 = y12;
y13 = -1/0.2j;
y31 = y13;
y23 = -1/0.4j;
y32 = y23;

% Define the admittance matrix Ybus using the elements calculated above
Ybus = [y11 y12 y13; y21 y22 y23; y31 y32 y33];
```

- (b)** Determine the bus impedance matrix Z_{bus} .

```
Zbus = inv(Ybus);
```

- (c)** Determine the voltage in all buses of the system. Please remember that in order to obtain the current injection vector I_{bus} you must convert the voltage sources into current sources using Norton's theorem



Source: [2]

where $I_{no} = \frac{V_{th}}{R_{th}}$ and $R_{no} = R_{th}$

```
% Define the voltage sources at buses 1 and 2
V1 = 1.28;
V2 = 1.20*exp(deg2rad(30)*1j);

% Define the equivalent current sources at buses 1 and 2 using Norton's
% theorem
I1 = V1/0.8j;
I2 = V2/1j;

% Define the current injection vector Ibus
Ibus = [I1; I2; 0];

% Calculate the bus voltage vector Vbus
Vbus = Zbus*Ibus; % Alternatively, Vbus = Ybus\Ibus;
```

(d) Calculate the complex power S supplied by each generator. Please remember that

$$S = V \cdot I^* = \frac{|V|^2}{Z^*} = |I|^2 \cdot Z$$

```
S1 = V1*conj((V1-Vbus(1))/0.8j);
S2 = V2*conj((V2-Vbus(2))/1j);
```

Laboratory 4: The Gauss-Seidel method (Cont.)

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

The power flow problem

The power flow equations are

$$P_i - jQ_i = V_i^* \sum_{k=1}^N Y_{ik} V_k$$

$$P_i = P_{gi} - P_{di}$$

$$Q_i = Q_{gi} - Q_{di}$$

Bus types are defined according to the quantities ($|V_i|$, δ_i , P_i , Q_i) specified for each bus, according to the table below.

Bus type	No. of buses	Quantities specified	No. of available equations	No. of δ_i , $ V_i $ state variables
Slack: $i = 1$	1	$\delta_1, V_1 $	0	0
Voltage controlled ($i = 2, \dots, N_g + 1$)	N_g	$P_i, V_i $	N_g	N_g
Load ($i = N_g + 2, \dots, N$)	$N - N_g - 1$	P_i, Q_i	$2(N - N_g - 1)$	$2(N - N_g - 1)$
Totals	N	$2N$	$2N - N_g - 2$	$2N - N_g - 2$

Source: [1]

Gauss-Seidel method

- **Step 1:** Define the type of each bus (load, voltage-controlled or slack bus) based on the variables known ($|V_i|$, δ_i , P_i , Q_i)
- **Step 2:** Build the admittance matrix Y_{bus}
- **Step 3:** Initialize load bus voltages (flat start \Rightarrow all voltages at $1.0\angle0^\circ$)
- **Step 4:** Define scheduled powers $P_{i,\text{sch}}$ and $Q_{i,\text{sch}}$
- **Step 5:** For each bus i ,

If i is **slack bus**, do nothing.

If i is load bus, calculate the voltage magnitude $|V_i|$ and angle δ_i using:

$$V_i^{(r)} = \frac{1}{Y_{ii}} \left\{ \frac{P_{i,sch} - jQ_{i,sch}}{V_i^{(r-1)*}} - \left[\sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} + \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

where $P_{i,sch} = P_{gi} - P_{di}$ and $Q_{i,sch} = Q_{gi} - Q_{di}$.

If using an acceleration factor α , then

$$V_{i,acc}^{(r)} = V_i^{(r-1)} + \alpha (V_i^{(r)} - V_i^{(r-1)})$$

If i is voltage-controlled bus, calculate the reactive power Q_i and voltage angle δ_i using:

$$Q_i^{(r)} = -Im \left\{ V_i^{(r-1)*} \left[\sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} + \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

$$\delta_i^{(r)} = \angle \left\{ \frac{1}{Y_{ii}} \left[\frac{P_{i,sch} - jQ_i}{V_i^{(r-1)*}} - \sum_{k=1}^{i-1} Y_{ik} V_k^{(r)} - \sum_{k=i+1}^N Y_{ik} V_k^{(r-1)} \right] \right\}$$

If $Q_i^{(r)} \leq Q_{i,min}$, set $Q_i^{(r)} = Q_{i,min}$ and treat the bus as a load bus.

If $Q_i^{(r)} \geq Q_{i,max}$, set $Q_i^{(r)} = Q_{i,max}$ and treat the bus as a load bus.

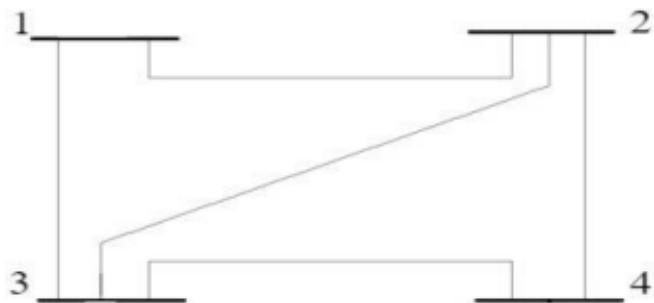
- **Step 6:** Check if algorithm converged using $|V_i^{(r)} - V_i^{(r-1)}| \leq \varepsilon$ and $|\delta_i^{(r)} - \delta_i^{(r-1)}| \leq \varepsilon$

If converged \Rightarrow calculate the power P and Q for the slack bus.

If not converged \Rightarrow repeat **Steps 5 and 6**.

Exercise 1: When PV buses are absent

Use the system below to answer the questions.



$$Y_{bus} = \begin{bmatrix} 3 - 9j & -2 + 6j & -1 + 3j & 0 \\ -2 + 6j & 3.66 - 11j & -0.66 + 2j & -1 + 3j \\ -1 + 3j & -0.66 + 2j & 3.66 - 11j & -2 + 6j \\ 0 & -1 + 3j & -2 + 6j & 3 - 9j \end{bmatrix}$$

Bus	P_{di}	Q_{di}	$ V_i $	δ_i	Bus type
1			1.04	0°	Slack
2	0.5	-0.2			PQ
3	-0.1	0.5			PQ
4	0.3	-0.1			PQ

Source: [2]

(a) Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 3; 3; 3];
```

(b) Determine the bus admittance matrix Y_{bus} .

```
Ybus = [3-9j -2+6j -1+3j 0;
       -2+6j 3.66-11j -0.66+2j -1+3j;
       -1+3j -0.66+2j 3.66-11j -2+6j;
       0 -1+3j -2+6j 3-9j];
```

(c) Define the voltage vector V_{bus} and initialize the load bus voltages.

```
Vinit = [1.04; 1; 1; 1];
```

(d) Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
P_sch = [inf; -0.5; 0.1; -0.3];

% Define the reactive power vector Q_sch
Q_sch = [inf; 0.2; -0.5; 0.1];
```

(e) Edit the file **GaussSeidel.mlx** to calculate the load bus voltages.

(f) Calculate the voltages V_{bus} in the buses using the GaussSeidel function. Use a tolerance of $\epsilon = 10^{-6}$.

```
% Define the tolerance
tolerance = 1e-6;

% Call the GaussSeidel function using the values defined previously and the
% tolerance
```

```
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance, 1);
```

(g) Use the voltages vector V_{bus} below to initialize the load bus voltages. Using the same tolerance as before, what happens to the results?

```
% Initial voltage guesses
Vinit_alt = [1.04; 4; 2; 3];

% Call the GaussSeidel function
GaussSeidel(Ybus, Vinit_alt, P_sch, Q_sch, bus_types, tolerance, 1);
```

(h) How many iterations the method takes to converge if you use a tolerance of $\varepsilon = 10^{-3}$? Repeat the analysis for $\varepsilon = 10^{-9}$.

```
% Call the GaussSeidel function with tolerance value 1e-3
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-3, 1);

% Call the GaussSeidel function with tolerance value 1e-9
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-9, 1);
```

(i) Calculate the powers P_1 and Q_1 in the slack bus. Please note that the power injected at bus 1 is given by

$$P_1 - jQ_1 = V_1^* \sum_{k=1}^N Y_{1k} V_k$$

```
% Calculate the voltages at the buses using the GaussSeidel function
Vbus = GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-9, 1);
% Calculate P1 - jQ1
S1 = conj(Vbus(1))*Ybus(1,:)*Vbus;

% Calculate active and reactive powers P1 and Q1
P1 = real(S1)
Q1 = -imag(S1)
```

(j) Calculate the total transmission losses. Please note that the transmission losses P_L and Q_L is given by

$$P_L = \sum_{i=1}^N P_i = \sum_{i=1}^N P_{gi} - \sum_{i=1}^N P_{di}$$

$$Q_L = \sum_{i=1}^N Q_i = \sum_{i=1}^N Q_{gi} - \sum_{i=1}^N Q_{di}$$

```
% Calculate the voltages at the buses using the GaussSeidel function
Vbus = GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-9, 1);
% Calculate the total power loss
PL = 0;
QL = 0;
for i = 1:4
    S = conj(Vbus(i))*Ybus(i,:)*Vbus;
    P = real(S);
```

```

Q = -imag(S);
PL = PL + P;
QL = QL + Q;
end

% Alternative solution using matrices instead of a for loop
S = transpose(conj(Vbus))*Ybus*Vbus;
PL = real(S);
QL = -imag(S);

```

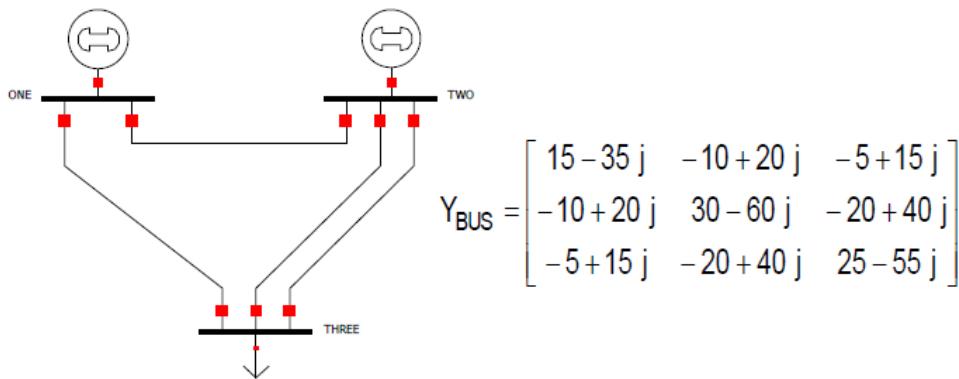
(k) Edit the file `GaussSeidel.mlx` to add an acceleration factor to the calculation of the bus voltages.

(l) Run the same experiment using an acceleration factor α . What happens if you use $\alpha = 2$?

```
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-6, 2);
```

Exercise 2: When PV buses are present

Use the system below to answer the questions.



Bus	P_{gi}	P_{di}	Q_{di}	$ V_i $	δ_i	Bus type
1				1.02	0°	Slack
2	0.5			1.02		PV
3		1.0	0.6			PQ

Source: [3]

(a) Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 2; 3];
```

(b) Determine the bus admittance matrix Y_{bus} .

```

Ybus = [15-35j -10+20j -5+15j;
        -10+20j 30-60j -20+40j;
        -5+15j -20+40j 25-55j];

```

(c) Define the voltage vector V_{bus} and initialize the load bus voltages.

```
Vinit = [1.2; 1.2; 1];
```

(d) Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch  
P_sch = [inf; 0.5; -1];  
  
% Define the reactive power vector Q_sch  
Q_sch = [inf; 0; -0.6];
```

(e) Edit the file `GaussSeidel mlx` to calculate the voltage-controlled bus reactive power and voltage angle.

(f) Calculate the voltages V_{bus} in the buses using the GaussSeidel function. Use a tolerance of $\varepsilon = 10^{-6}$.

```
% Define the tolerance  
tolerance = 1e-6;  
  
% Call the GaussSeidel function using the values defined previously and the  
% tolerance  
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-6, 1.2);
```

(g) Calculate the powers P_1 and Q_1 in the slack bus.

```
% Calculate the voltages at the buses using the GaussSeidel function  
Vbus = GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-9, 1);  
  
% Calculate P1 - jQ1  
S1 = conj(Vbus(1))*Ybus(1,:)*Vbus;  
  
% Calculate active and reactive powers P1 and Q1  
P1 = real(S1)  
Q1 = -imag(S1)
```

(h) Calculate the total transmission losses P_L and Q_L .

```
% Calculate the voltages at the buses using the GaussSeidel function  
Vbus = GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, 1e-9, 1);  
  
% Calculate the loss PL  
S = transpose(conj(Vbus))*Ybus*Vbus;  
PL = real(S)  
QL = -imag(S)
```

```

function V = GaussSeidel(Y, V, P, Q, busTypes, tolerance, acceleration)
    % Save results for printing
    Viter(:,1) = V;

    % Total number of buses
    nbuses = size(Y,1);

    tStart=tic; % start the iteration time clock

    error=1;
    iteration=1;
    Vprev=V;

    % The algorithm runs until the stop condition is met
    while (error > tolerance && iteration < 500)

        % Iterate over each bus.
        for i=1:nbuses
            % If slack bus
            if busTypes(i) == 1
                continue
            end

            % Calculate the summation term over Vk * Yik
            PYV=0;
            for k=1:nbuses
                if k ~= i
                    % Calculate Vk * Yik and add it to the total
                    PYV = PYV + Y(i,k)* V(k);
                end
            end

            % If bus is PV
            if busTypes(i) == 2
                % Estimate Qi at each iteration for the PV buses
                Q(i) = -imag(conj(V(i))*(PYV + Y(i,i)*V(i)));
            end

            % Calculate the bus voltage
            V(i) = (1/Y(i,i))*((P(i)-1j*Q(i))/conj(V(i))-PYV);

            % Calculate the bus voltage considering the acceleration factor
            V(i) = Vprev(i) + acceleration*(V(i) - Vprev(i));

            % If bus is PV
            if busTypes(i) == 2
                % Calculate the bus voltage for the PV bus
                V(i) = abs(Vprev(i))*exp(angle(V(i))*1j);
            end
        end
    end

```

```

% Tolerance at the current iteration
error_mag = max(abs(abs(V) - abs(Vprev)));
error_angle = max(abs(angle(V) - angle(Vprev)));
error = max(error_mag, error_angle);

% Save current voltage vector for next iteration
Vprev=V;

% Save results for printing
Viter(:,iteration+1)=V;

% Increment iteration count
iteration = iteration + 1;
end

% +++++++ Print results ++++++
disp('                                     Gauss Seidel Load-Flow Study')
disp('                                     Report of Power Flow Calculations ')
disp(' ')
fprintf('Number of iterations      : %g \n', iteration)
fprintf('Solution time            : %02.5g sec.\n', toc(tStart))
disp('Iteration    Voltages')
for i=1:size(Viter,2)
    fprintf('    %02d      ', i-1)
    for k=1:size(Viter,1)
        fprintf('%02.3f < %02.3f      ', abs(Viter(k,i)), rad2deg(angle(Viter(k,i))))
    end
    fprintf('\n')
end
end

```

Laboratory 6: Newton-Raphson and Fast Decoupled methods

Course: EES5030 Energiasüsteemide optimaaljuhtimine

Author: Nathalia Campos

The power flow problem

The Newton-Raphson method calculates the bus voltages in the network using the polar form of the net active P_i and reactive Q_i powers transmitted in the grid:

$$P_i = |V_i|^2 G_{ii} + \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_i = -|V_i|^2 B_{ii} - \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

The values of the bus voltages are estimated such as to minimize the power mismatches ΔP_i and ΔQ_i :

$$\Delta P_i = P_{i,sch} - P_{i,calc}$$

$$\Delta Q_i = Q_{i,sch} - Q_{i,calc}$$

This is achieved by calculating the sensitivity of active and reactive powers in the transmission lines to changes in the voltage magnitudes and angles. This information is represented by the Jacobian and is used to make an educated decision on how to correct the voltage guesses.

$$\begin{bmatrix} \Delta \mathbf{P}_i \\ \Delta \mathbf{Q}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{P}}{\partial |\mathbf{V}|} \\ \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{Q}}{\partial |\mathbf{V}|} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\delta} \\ \Delta |\mathbf{V}| \end{bmatrix}$$

The Newton-Raphson method

Steps to implement the Newton-Raphson method:

- **Step 1:** Define the type of each bus (load, voltage-controlled or slack bus) based on the variables known ($|V_i|$, δ_i , P_i , Q_i)
- **Step 2:** Build the admittance matrix Y_{bus}
- **Step 3:** Initialize bus voltages (flat start \Rightarrow all voltages at $1.0\angle 0^\circ$)
- **Step 4:** Define scheduled powers $P_{i,sch}$ and $Q_{i,sch}$
- **Step 5:** Calculate the net transmitted powers $P_{i,calc}$ and $Q_{i,calc}$

$$P_{i,calc} = |V_i|^2 G_{ii} + \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_{i,calc} = -|V_i|^2 B_{ii} - \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

where $Y_{ik} = G_{ik} + jB_{ik}$ and $Y_{ik} = |Y_{ik}| \angle \theta_{ik}$.

- **Step 6:** Calculate the power mismatches ΔP_i and ΔQ_i

$$\Delta P_i = P_{i,sch} - P_{i,calc}$$

$$\Delta Q_i = Q_{i,sch} - Q_{i,calc}$$

- **Step 7:** Build the Jacobian matrix \mathbf{J}
- **Step 8:** Find the vector with magnitude $\Delta|V_i|$ and angle $\Delta\delta_i$ corrections:

$$\begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta|V| \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \Delta\mathbf{P}_i \\ \Delta\mathbf{Q}_i \end{bmatrix}$$

- **Step 9:** Calculate the corrected values for the voltage magnitudes and angles

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

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

- **Step 10:** Check if algorithm converged using $|V_i^{(r)} - V_i^{(r-1)}| \leq \varepsilon$ and $|\delta_i^{(r)} - \delta_i^{(r-1)}| \leq \varepsilon$

If converged \Rightarrow calculate the power P and Q for the slack bus.

If not converged \Rightarrow repeat **Step 5 through 10**.

How to build the Jacobian \mathbf{J}

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 \\ \mathbf{J}_3 & \mathbf{J}_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{P}}{\partial |V|} \\ \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \mathbf{Q}}{\partial |V|} \end{bmatrix}$$

\mathbf{J}_1 elements

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial P_i}{\partial \delta_j} = -|V_i V_j Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

J₂ elements

$$\frac{\partial P_i}{\partial V_i} = 2|V_i Y_{ii}| \cos \theta_{ii} + \sum_{k=1, k \neq i}^N |V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial P_i}{\partial V_j} = |V_i Y_{ij}| \cos(\theta_{ij} + \delta_j - \delta_i)$$

J₃ elements

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{k=1, k \neq i}^N |V_i V_k Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial Q_i}{\partial \delta_j} = -|V_i V_j Y_{ij}| \cos(\theta_{ij} + \delta_j - \delta_i)$$

J₄ elements

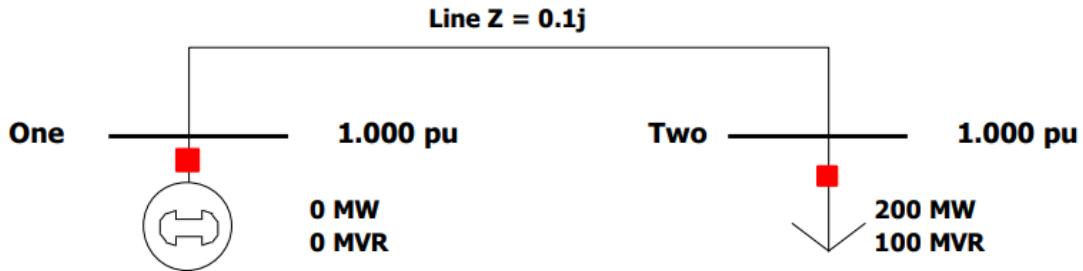
$$\frac{\partial Q_i}{\partial V_i} = -2|V_i Y_{ii}| \sin \theta_{ii} - \sum_{k=1, k \neq i}^N |V_k Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

$$\frac{\partial Q_i}{\partial V_j} = -|V_i Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

Exercise 1

Use the system below to answer the questions. Assume that $S_{\text{base}} = 100\text{MVA}$ and note that the active P_{pu} and reactive Q_{pu} power in p.u. can be obtained using

$$P_{pu} = \frac{P_{MW}}{S_{\text{base},MVA}} \text{ and } Q_{pu} = \frac{Q_{Mvar}}{S_{\text{base},MVA}}$$



- (a)** Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 3];
```

- (b)** Determine the bus admittance matrix Y_{bus} .

```
% Define the diagonal elements of Ybus
y11 = 1/0.1j;
y22 = y11;

% Define the off-diagonal elements of Ybus
y12 = -1/0.1j;
y21 = y12;

% Define the admittance matrix Ybus using the elements calculated above
Ybus = [y11 y12; y21 y22];
```

- (c)** Define the voltage vector V_{init} and initialize the bus voltages.

```
Vinit = [1; 1];
```

- (d)** Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
P_sch = [inf; -200/100];

% Define the reactive power vector Q_sch
Q_sch = [inf; -100/100];
```

- (e)** Edit the file **NewtonRaphson.mlx** to calculate the bus voltages.

- (f)** Calculate the voltages V_{bus} in the buses using the **NewtonRaphson** function. Use a tolerance of $\varepsilon = 10^{-12}$.

```
% Define the tolerance
tolerance = 1e-12;

% Call the NewtonRaphson function using the values defined previously and the
% tolerance
```

```
NewtonRaphson(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance);
```

Newton-Raphson Load-Flow Study
Report of Power Flow Calculations

Number of iterations : 7
Solution time : 0.0048556 sec.
Iteration Voltages
00 1.000 ∠ 0.000 1.000 ∠ 0.000
01 1.000 ∠ 0.000 0.900 ∠ -11.459
02 1.000 ∠ 0.000 0.859 ∠ -13.370
03 1.000 ∠ 0.000 0.855 ∠ -13.521
04 1.000 ∠ 0.000 0.855 ∠ -13.522
05 1.000 ∠ 0.000 0.855 ∠ -13.522
06 1.000 ∠ 0.000 0.855 ∠ -13.522

(g) Repeat the calculation with the GaussSeidel function from the previous lab using the same parameters. Are there any noticeable differences?

```
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance, 1);
```

Gauss Seidel Load-Flow Study
Report of Power Flow Calculations

Number of iterations : 24
Solution time : 0.0013422 sec.
Iteration Voltages
00 1.000 ∠ 0.000 1.000 ∠ 0.000
01 1.000 ∠ 0.000 0.922 ∠ -12.529
02 1.000 ∠ 0.000 0.868 ∠ -12.529
03 1.000 ∠ 0.000 0.861 ∠ -13.431
04 1.000 ∠ 0.000 0.857 ∠ -13.431
05 1.000 ∠ 0.000 0.856 ∠ -13.513
06 1.000 ∠ 0.000 0.855 ∠ -13.513
07 1.000 ∠ 0.000 0.855 ∠ -13.521
08 1.000 ∠ 0.000 0.855 ∠ -13.521
09 1.000 ∠ 0.000 0.855 ∠ -13.522
10 1.000 ∠ 0.000 0.855 ∠ -13.522
11 1.000 ∠ 0.000 0.855 ∠ -13.522
12 1.000 ∠ 0.000 0.855 ∠ -13.522
13 1.000 ∠ 0.000 0.855 ∠ -13.522
14 1.000 ∠ 0.000 0.855 ∠ -13.522
15 1.000 ∠ 0.000 0.855 ∠ -13.522
16 1.000 ∠ 0.000 0.855 ∠ -13.522
17 1.000 ∠ 0.000 0.855 ∠ -13.522
18 1.000 ∠ 0.000 0.855 ∠ -13.522
19 1.000 ∠ 0.000 0.855 ∠ -13.522
20 1.000 ∠ 0.000 0.855 ∠ -13.522
21 1.000 ∠ 0.000 0.855 ∠ -13.522
22 1.000 ∠ 0.000 0.855 ∠ -13.522
23 1.000 ∠ 0.000 0.855 ∠ -13.522

The Fast Decoupled method

Simplifications:

- Strong relationship between $\Delta\delta \rightarrow \Delta P$ and $\Delta|V| \rightarrow \Delta Q$. This implies $\mathbf{J}_2 = \mathbf{0}$ and $\mathbf{J}_3 = \mathbf{0}$.
- Small angle difference between buses, $\delta_j \approx \delta_i$

- In a transmission line, $R \ll X$

The corrections vector is then written as:

$$[\Delta\delta] = J_1^{-1} [\Delta P]$$

$$[\Delta|V|] = J_4^{-1} [\Delta Q]$$

How to build the Jacobian J

$$J = \begin{bmatrix} J_1 & \mathbf{0} \\ \mathbf{0} & J_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \mathbf{0} \\ \mathbf{0} & \frac{\partial Q}{\partial |V|} \end{bmatrix}$$

J_1 elements

$$\frac{\partial P_i}{\partial \delta_i} = -|V_i|^2 B_{ii}$$

$$\frac{\partial P_i}{\partial \delta_j} = -|V_i V_j| B_{ij}$$

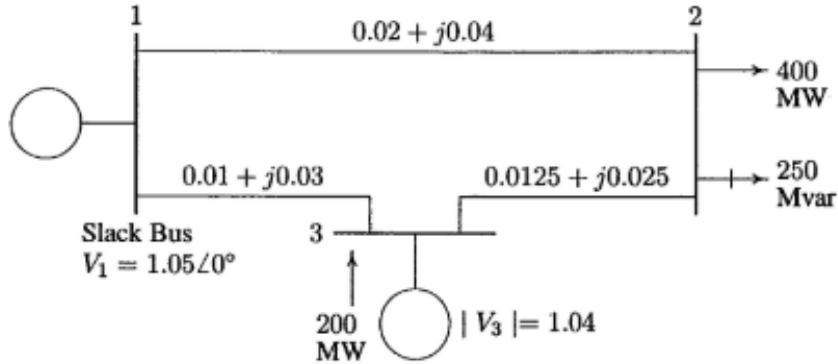
J_4 elements

$$\frac{\partial Q_i}{\partial V_i} = -|V_i| B_{ii}$$

$$\frac{\partial Q_i}{\partial V_j} = -|V_i| B_{ij}$$

Exercise 2

Use the system below to answer the questions. Assume that $S_{\text{base}} = 100\text{MVA}$.



- (a)** Define a vector of bus types. Use the following code: 1 = slack bus, 2 = voltage-controlled bus, 3 = load bus.

```
bus_types = [1; 3; 2];
```

- (b)** Determine the bus admittance matrix Y_{bus} .

```
y_t12 = 1/(0.02+0.04j);
y_t13 = 1/(0.01+0.03j);
y_t23 = 1/(0.0125+0.025j);

% Define the diagonal elements of Ybus
y11 = y_t12 + y_t13;
y22 = y_t23 + y_t12;
y33 = y_t23 + y_t13;

% Define the off-diagonal elements of Ybus
y12 = -y_t12;
y21 = y12;
y13 = -y_t13;
y31 = y13;
y23 = -y_t23;
y32 = y23;

% Define the admittance matrix Ybus using the elements calculated above
Ybus = [y11 y12 y13; y21 y22 y23; y31 y32 y33];
```

- (c)** Define the voltage vector V_{init} and initialize the bus voltages.

```
Vinit = [1.05; 1; 1.04];
```

- (d)** Define the vectors for active P_{sch} and reactive Q_{sch} power scheduled at the buses.

```
% Define the active power vector P_sch
P_sch = [inf; -400/100; 200/100];

% Define the reactive power vector Q_sch
Q_sch = [inf; -250/100; 0];
```

- (e)** Save the **NewtonRaphson.mlx** file as **FastDecoupled.mlx** and implement the necessary changes to calculate the bus voltages using the Fast Decoupled method.

(f) Calculate the voltages V_{bus} in the buses using the `FastDecoupled` function. Use a tolerance of $\epsilon = 10^{-12}$.

```
FastDecoupled(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance);
```

Fast Decoupled Load-Flow Study
Report of Power Flow Calculations

Number of iterations	:	37				
Solution time	:	0.0025364 sec.				
Iteration		Voltages				
00	1.050	$\angle 0.000$	1.000	$\angle 0.000$	1.040	$\angle 0.000$
01	1.050	$\angle 0.000$	0.996	$\angle -3.465$	1.040	$\angle -0.491$
02	1.050	$\angle 0.000$	0.965	$\angle -3.236$	1.040	$\angle -0.458$
03	1.050	$\angle 0.000$	0.966	$\angle -2.504$	1.040	$\angle -0.497$
04	1.050	$\angle 0.000$	0.973	$\angle -2.571$	1.040	$\angle -0.515$
05	1.050	$\angle 0.000$	0.973	$\angle -2.742$	1.040	$\angle -0.499$
06	1.050	$\angle 0.000$	0.971	$\angle -2.727$	1.040	$\angle -0.495$
07	1.050	$\angle 0.000$	0.971	$\angle -2.686$	1.040	$\angle -0.499$
08	1.050	$\angle 0.000$	0.972	$\angle -2.689$	1.040	$\angle -0.500$
09	1.050	$\angle 0.000$	0.972	$\angle -2.699$	1.040	$\angle -0.499$
10	1.050	$\angle 0.000$	0.972	$\angle -2.698$	1.040	$\angle -0.499$
11	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
12	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
13	1.050	$\angle 0.000$	0.972	$\angle -2.697$	1.040	$\angle -0.499$
14	1.050	$\angle 0.000$	0.972	$\angle -2.697$	1.040	$\angle -0.499$
15	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
16	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
17	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
18	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
19	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
20	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
21	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
22	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
23	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
24	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
25	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
26	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
27	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
28	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
29	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
30	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
31	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
32	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
33	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
34	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
35	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
36	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$

(g) Compare the results with the outputs from the `GaussSeidel` and `NewtonRaphson` functions.

```
Vbus = NewtonRaphson(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance);
```

Newton-Raphson Load-Flow Study
Report of Power Flow Calculations

Number of iterations	:	6				
Solution time	:	0.012734 sec.				
Iteration		Voltages				
00	1.050	$\angle 0.000$	1.000	$\angle 0.000$	1.040	$\angle 0.000$
01	1.050	$\angle 0.000$	0.973	$\angle -2.593$	1.040	$\angle -0.442$
02	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$
03	1.050	$\angle 0.000$	0.972	$\angle -2.696$	1.040	$\angle -0.499$

04	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$
05	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$

```
GaussSeidel(Ybus, Vinit, P_sch, Q_sch, bus_types, tolerance, 1);
```

Gauss Seidel Load-Flow Study
Report of Power Flow Calculations

Iteration	Voltages			
00	$1.050 \angle 0.000$	$1.000 \angle 0.000$	$1.040 \angle 0.000$	
01	$1.050 \angle 0.000$	$0.976 \angle -2.486$	$1.040 \angle -0.285$	
02	$1.050 \angle 0.000$	$0.972 \angle -2.561$	$1.040 \angle -0.403$	
03	$1.050 \angle 0.000$	$0.972 \angle -2.642$	$1.040 \angle -0.459$	
04	$1.050 \angle 0.000$	$0.972 \angle -2.674$	$1.040 \angle -0.482$	
05	$1.050 \angle 0.000$	$0.972 \angle -2.687$	$1.040 \angle -0.492$	
06	$1.050 \angle 0.000$	$0.972 \angle -2.693$	$1.040 \angle -0.496$	
07	$1.050 \angle 0.000$	$0.972 \angle -2.695$	$1.040 \angle -0.498$	
08	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.498$	
09	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
10	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
11	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
12	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
13	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
14	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
15	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
16	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
17	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
18	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
19	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
20	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
21	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
22	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
23	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
24	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
25	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
26	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	
27	$1.050 \angle 0.000$	$0.972 \angle -2.696$	$1.040 \angle -0.499$	

(h) Determine the current in each transmission line. Please remember that the current between buses i and j flowing in a transmission line having an impedance Z_{line} is given by

$$I = \frac{V_i - V_j}{Z_{\text{line}}} = (V_i - V_j) \cdot Y_{\text{line}}$$

```
% Calculate the current for each line
I12 = (Vbus(1)-Vbus(2))*y_t112;
I13 = (Vbus(1)-Vbus(3))*y_t113;
I23 = (Vbus(2)-Vbus(3))*y_t123;

% Create a column vector Ilines with the line currents
Ilines = [I12; I13; I23];
```

(i) Determine the loading factor for each line. Are any of the lines overloaded? Please note that the loading factor is given by

$$Loading = \frac{|I|}{I_{rating}}$$

$$Loading, \% = Loading \times 100$$

```
% Current ratings for each line in p.u.
Irating12 = 2;
Irating13 = 2;
Irating23 = 2;

% Create a column vector Irating with the line ratings
% Follow the same order as Ilines
Irating = [Irating12; Irating13; Irating23];

% Calculate the loading
loading = abs(Ilines)./Irating;
loading_percent = loading * 100;
```

- (j) Assuming all voltages must be between 0.95 and 1.10 p.u., determine whether the system has any high or low voltage buses.

No, all bus voltages are within the voltage limits.

```

function V = NewtonRaphsonSolution(Y, V, P, Q, bus_types, tolerance)
DEBUG = false;

% Save results for printing
Viter(:,1) = V;

% Total number of buses
nbuses = size(Y,1);

% Total number of PV busses
nbuses_pv = sum(bus_types==2);

% Total number of PQ busses
nbuses_pq = sum(bus_types==3);

tStart=tic; % start the iteration time clock

error=1;
iteration=1;
Vprev=V;

% The algorithm runs until the stop condition is met
while (error > tolerance && iteration < 500)

%----- Delta P -----%
for i = 2:nbuses
VVY = 0;
for k = 1:nbuses
if i~=k
% Calculate the VVY product
VVY = VVY + abs(Y(i,k))*V(k)*V(i))*cos(angle(Y(i,k))+...
angle(V(k))-angle(V(i))); % multiplying admittance & voltage
end
end

% Calculate Pcalc(i)
Pcalc(i) = ((abs(V(i))^2)*real(Y(i,i)))+VVY;

% Calculate the power mismatch deltaP(i)
deltaP(i) = P(i)-Pcalc(i);

end

if (iteration == 1 && DEBUG)
    deltaP
end

%----- Delta Q -----%

```

```

for i = 2:nbuses
    VVY = 0;
    for k = 1:nbuses
        if i~=k
            % Calculate the VVY product
            VVY = VVY + abs(Y(i,k)*V(k)*V(i))*sin(angle(Y(i,k))+...
                angle(V(k))-angle(V(i))); % multiplying admittance & voltage
        end
    end

    % Calculate Qcalc(i)
    Qcalc(i) = -((abs(V(i))^2)*imag(Y(i,i)))-VVY;

    % Calculate the power mismatch deltaQ(i)
    deltaQ(i) = Q(i)-Qcalc(i);
end

% Deletes rows with PV buses
pv_indices = bus_types==2;
deltaQ(pv_indices) = [];

if (iteration == 1 && DEBUG)
    deltaQ
end

%----- P and Q mismatches -----
mismatches = [deltaP(2:end) deltaQ(2:end)];

%----- Jacobian -----
J1=zeros(nbuses-1,nbuses-1);
J2=zeros(nbuses-1,nbuses_pq);
J3=zeros(nbuses_pq,nbuses-1);
J4=zeros(nbuses_pq,nbuses_pq);

%----- J1 -----
for i=2:nbuses
    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J1(i-1,j-1)
            J1(i-1,j-1) = -abs(Y(i,j)*V(i)*V(j))*sin(angle(Y(i,j))+...
                angle(V(j))-angle(V(i)));
        end

        % Diagonal elements
        VVY=0;
    end
end

```

```

if i==j
    for k=1:nbuses
        if k~=i
            % Calculate the VVY product
            VVY = VVY + abs(Y(i,k)*V(i)*V(k))*sin(angle(Y(i,k))+...
                angle(V(k))-angle(V(i)));
        end
    end
    % Calculate the element J1(i-1,j-1)
    J1(i-1,j-1) = VVY;
end
end

if (iteration == 1 && DEBUG)
    J1
end

%-----J2-----%
for i=2:nbuses
    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J2(i-1,j-1)
            J2(i-1,j-1) = abs(Y(i,j)*V(i))*cos(angle(Y(i,j))+...
                angle(V(j))-angle(V(i)));
        end

        % Diagonal elements
        VY=0;
        if i==j
            for k=1:nbuses
                if k~=i
                    % Calculate the VY product
                    VY = VY + abs(Y(i,k)*V(k))*cos(angle(Y(i,k))+...
                        angle(V(k))-angle(V(i)));
                end
            end
            % Calculate the element J2(i-1,j-1)
            J2(i-1,j-1) = 2*abs(Y(i,i)*V(i))*cos(angle(Y(i,i))) + VY;
        end
    end
end

% Deletes columns associated with PV buses
pv_indices = bus_types(2:end,:)==2;
J2(:,pv_indices) = [];

if (iteration == 1 && DEBUG)

```

```

J2
end

%-----J3-----%
for i=2:nbuses
    for j=2:nbuses

        % Off-diagonal elements
        if i~=j
            % Calculate the element J3(i-1,j-1)
            J3(i-1,j-1) = -abs(Y(i,j)*V(i)*V(j))*cos(angle(Y(i,j))+...
                angle(V(j))-angle(V(i)));
        end

        % Diagonal elements
        VVY=0;
        if i==j
            for k=1:nbuses
                if k~=i
                    % Calculate the VY product
                    VVY = VVY + abs(Y(i,k)*V(i)*V(k))*cos(angle(Y(i,k))+...
                        angle(V(k))-angle(V(i)));
                end
            end
            % Calculate the element J3(i-1,j-1)
            J3(i-1,j-1) = VVY;
        end
    end
end

% Deletes rows associated with PV buses
pv_indices = bus_types(2:end,:)==2;
J3(pv_indices,:) = [];

if (iteration == 1 && DEBUG)
    J3
end

%-----J4-----%
for i=2:nbuses

    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J4(i-1,j-1)
            J4(i-1,j-1) = -abs(Y(i,j)*V(i))*sin(angle(Y(i,j))+...

```

```

        angle(V(j))-angle(V(i)));
    end

    % Diagonal elements
    VY=0;
    if i==j
        for k=1:nbuses
            if k~=i
                % Calculate the VY product
                VY = VY + abs(V(k)*Y(i,k))*sin(angle(Y(i,k))+...
                    angle(V(k))-angle(V(i)));
            end
        end
        % Calculate the element J4(i-1,j-1)
        J4(i-1,j-1) = -2*abs(V(i)*Y(i,i))*sin(angle(Y(i,i))) - VY;
    end
end

% Deletes rows and columns associated with PV buses
pv_indices = bus_types(2:end,:)==2;
J4(:,pv_indices) = [];
J4(pv_indices,:) = [];

if (iteration == 1 && DEBUG)
    J4
end

J = [J1 J2; J3 J4];

%-----Corrections vector-----

% Calculate the corrections vector corrections
corrections=J\mismatches';

if (iteration == 1 && DEBUG)
    corrections
end

%-----New estimates-----

for k=2:nbuses
    new_angle = angle(V(k)) + corrections(k-1);
    V(k) = abs(V(k))*exp(new_angle*1j);
end

pv_bus_count = 0;
for k=2:nbuses
    if bus_types(k) ~= 2
        new_mag = abs(V(k)) + corrections(nbuses-2+k-pv_bus_count);
    end

```

```

        V(k) = new_mag*exp(angle(V(k))*1j);
    else
        pv_bus_count = pv_bus_count + 1;
    end
end

%-----

% Tolerance at the current iteration
error_mag = max(abs(abs(V) - abs(Vprev)));
error_angle = max(abs(angle(V) - angle(Vprev)));
error = max(error_mag, error_angle);

% Save current voltage vector for next iteration
Vprev=V;

% Save results for printing
Viter(:,iteration+1)=V;

% Increment iteration count
iteration=iteration+1;
end

% +++++++ Print results ++++++
disp('                                Newton-Raphson Load-Flow Study')
disp('                                Report of Power Flow Calculations ')
disp(' ')
fprintf('Number of iterations      : %g \n', iteration)
fprintf('Solution time            : %02.5g sec.\n',toc(tStart))
disp('Iteration      Voltages')
for i=1:size(Viter,2)
    fprintf('    %02d      ', i-1)
    for k=1:size(Viter,1)
        fprintf('%02.3f < %02.3f      ', abs(Viter(k,i)), rad2deg(angle(Viter(k,i))))
    end
    fprintf('\n')
end

end

```

Reference: Hafiz Kashif Khaleel (2022). Newton Raphson Power Flow Solution using MATLAB (<https://www.mathworks.com/matlabcentral/fileexchange/26391-newton-raphson-power-flow-solution-using-matlab>), MATLAB Central File Exchange. Retrieved September 26, 2022.

```

function V = FastDecoupled(Y, V, P, Q, bus_types, tolerance)
DEBUG = false;

% Save results for printing
Viter(:,1) = V;

% Total number of buses
nbuses = size(Y,1);

% Total number of PV busses
nbuses_pv = sum(bus_types==2);

% Total number of PQ busses
nbuses_pq = sum(bus_types==3);

tStart=tic; % start the iteration time clock

error=1;
iteration=1;
Vprev=V;

% The algorithm runs until the stop condition is met
while (error > tolerance && iteration < 500)

%----- Delta P -----%
for i = 2:nbuses
VVY = 0;
for k = 1:nbuses
if i~=k
% Calculate the VVY product
VVY = VVY + abs(Y(i,k))*V(k)*V(i))*cos(angle(Y(i,k))+...
angle(V(k))-angle(V(i))); % multiplying admittance & voltage
end
end

% Calculate Pcalc(i)
Pcalc(i) = ((abs(V(i))^2)*real(Y(i,i)))+VVY;

% Calculate the power mismatch deltaP(i)
deltaP(i) = P(i)-Pcalc(i);

end

if (iteration == 1 && DEBUG)
    deltaP
end

%----- Delta Q -----%

```

```

for i = 2:nbuses
    VVY = 0;
    for k = 1:nbuses
        if i~=k
            % Calculate the VVY product
            VVY = VVY + abs(Y(i,k)*V(k)*V(i))*sin(angle(Y(i,k))+...
                angle(V(k))-angle(V(i))); % multiplying admittance & voltage
        end
    end

    % Calculate Qcalc(i)
    Qcalc(i) = -((abs(V(i))^2)*imag(Y(i,i)))-VVY;

    % Calculate the power mismatch deltaQ(i)
    deltaQ(i) = Q(i)-Qcalc(i);
end

% Deletes rows with PV buses
pv_indices = bus_types==2;
deltaQ(pv_indices) = [];

if (iteration == 1 && DEBUG)
    deltaQ
end

%----- P and Q mismatches -----
mismatches_P = deltaP(2:end);
mismatches_Q = deltaQ(2:end);

%----- Jacobian-----
J1=zeros(nbuses-1,nbuses-1);
J2=zeros(nbuses-1,nbuses_pq);
J3=zeros(nbuses_pq,nbuses-1);
J4=zeros(nbuses_pq,nbuses_pq);

%----- J1 -----
for i=2:nbuses
    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J1(i-1,j-1)
            J1(i-1,j-1) = -abs(V(i)*V(j))*imag(Y(i,j));
        end

        % Diagonal elements
        if i==j

```

```

        % Calculate the element J1(i-1,j-1)
        J1(i-1,j-1) = -abs(V(i)*V(j))*imag(Y(i,j));
    end
end

if (iteration == 1 && DEBUG)
    J1
end

%-----J4-----%
for i=2:nbuses

    for j=2:nbuses
        % Off-diagonal elements
        if i~=j
            % Calculate the element J4(i-1,j-1)
            J4(i-1,j-1) = -abs(V(i))*imag(Y(i,j));;
        end

        % Diagonal elements
        if i==j
            % Calculate the element J4(i-1,j-1)
            J4(i-1,j-1) = -abs(V(i))*imag(Y(i,j));
        end
    end
end

% Deletes rows and columns associated with PV buses
pv_indices = bus_types(2:end,:)==2;
J4(:,pv_indices) = [];
J4(pv_indices,:) = [];

if (iteration == 1 && DEBUG)
    J4
end

%-----Corrections vector-----

% Calculate the corrections vector corrections
corrections_angle = J1\mismatches_P';
corrections_mag = J4\mismatches_Q';
corrections = [corrections_angle; corrections_mag];

if (iteration == 1 && DEBUG)
    corrections
end

```

```

%-----New estimates-----

for k=2:nbuses
    new_angle = angle(V(k)) + corrections(k-1);
    V(k) = abs(V(k))*exp(new_angle*1j);
end

pv_bus_count = 0;
for k=2:nbuses
    if bus_types(k) ~= 2
        new_mag = abs(V(k)) + corrections(nbuses-2+k-pv_bus_count);
        V(k) = new_mag*exp(angle(V(k))*1j);
    else
        pv_bus_count = pv_bus_count + 1;
    end
end

%

% Tolerance at the current iteration
error_mag = max(abs(abs(V) - abs(Vprev)));
error_angle = max(abs(angle(V) - angle(Vprev)));
error = max(error_mag, error_angle);

% Save current voltage vector for next iteration
Vprev=V;

% Save results for printing
Viter(:,iteration+1)=V;

% Increment iteration count
iteration=iteration+1;
end

% +++++++ Print results ++++++
disp(''Fast Decoupled Load-Flow Study')
disp(''Report of Power Flow Calculations ')
disp(' ')
fprintf('Number of iterations : %g \n', iteration)
fprintf('Solution time : %02.5g sec.\n', toc(tStart))
disp('Iteration Voltages')
for i=1:size(Viter,2)
    fprintf('%02d ', i-1)
    for k=1:size(Viter,1)
        fprintf('%02.3f < %02.3f ', abs(Viter(k,i)), rad2deg(angle(Viter(k,i))))
    end
    fprintf('\n')
end

end

```

Reference: Hafiz Kashif Khaleel (2022). Newton Raphson Power Flow Solution using MATLAB (<https://www.mathworks.com/matlabcentral/fileexchange/26391-newton-raphson-power-flow-solution-using-matlab>), MATLAB Central File Exchange. Retrieved September 26, 2022.

Kirjandus

- [1] M. Leinakse. *Elektrisüsteemi arvutamise alused reaalajasimulatsioonide raamistikus*, Tallinna Tehnikaülikool, 2021.
- [2] H. K. Khaleel (2022). Newton Raphson Power Flow Solution using MATLAB (<https://www.mathworks.com/matlabcentral/fileexchange/26391-newtonraphson-power-flow-solution-using-matlab>), MATLAB Central File Exchange. Retrieved September 26, 2022