

Prof. Silvan Antônio Flávio

1) Faça um programa computacional para solução do Fluxo de Potência por meio do método de Newton Raphson, seguindo o algoritmo descrito nas notas de aula (slide 56).

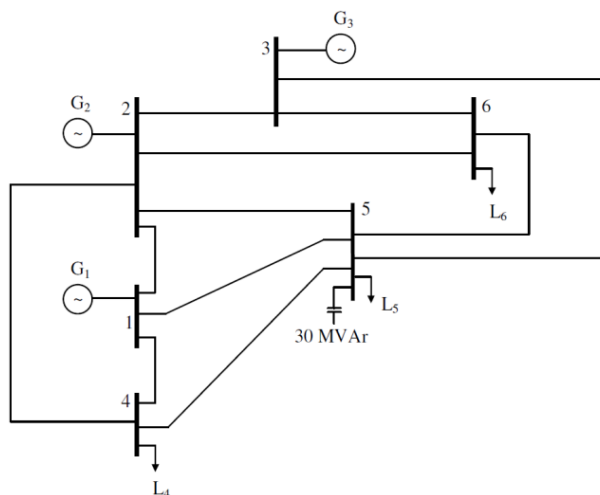
Observações:

- O programa deverá imprimir os resultados de barra e os fluxos nas linhas.
- Utilize o exemplo de 3 barras resolvido em sala de aula para testar o programa.

2) Utilize o programa para obter os resultados para o sistema abaixo. Faça uma análise do ponto de operação com e sem o banco de capacitor da barra 5.

Barra	Tipo	V (pu)	Ang. (graus)	P _G (MW)	Q _G (MVar)	P _D (MW)	Q _D (MVar)	Q _{sh} (MVar)
1						0	0	0
2						0	0	0
3						0	0	0
4						70	25	0
5						90	30	-30
6						86	35	0

De	Para	R (%)	X (%)	b (%)	tap (pu)	defas (pu)
1	2	4,0	20,0	2,0	0,000	0,000
1	4	4,0	20,0	2,0	0,000	0,000
1	5	6,0	30,0	3,0	0,000	0,000
2	3	5,0	25,0	2,0	0,000	0,000
2	4	2,0	10,0	1,0	0,000	0,000
2	5	6,0	30,0	3,0	0,000	0,000
2	6	3,5	20,0	2,0	0,000	0,000
3	5	5,0	26,0	3,0	0,000	0,000
3	6	2,0	10,0	1,0	0,000	0,000
4	5	8,0	40,0	4,0	0,000	0,000
5	6	6,0	30,0	3,0	0,000	0,000



Dicas para programação:

- Primeiramente, padronize a leitura dos dados de forma a facilitar o entendimento do programa. Siga a estrutura/ordem das tabelas acima, em que foram fornecidos os dados. Assim que fizer a leitura dos dados, faça uma interface com os nomes das variáveis. Por exemplo: `Tipo(:,1)=dados(:,2)` {e.g. 1 - Swing ; 2 - PV ; 3 - PQ }; `V(:,1)=dados(:,3)`; `Theta(:,1)=dados(:,4)*pi/180`; `Pg(:,1)=dados(:,5)`; e assim por diante. É importante que o algoritmo seja desenvolvido utilizando variáveis conhecidas da modelagem estudada em sala de aula.
- Defina o número de barras e circuitos, de acordo com os dados. Exemplo:
`nb=size(dados,1)` {número de barras}
- Em seguida, faça a montagem da Ybus e confira detalhadamente se está tudo certo, utilizando o exercício de três barras. Caso seja necessário, confira com seu colega de classe. Lembre-se que Geradores e Cargas não fazem parte da Ybus. Obtenha os parâmetros G e B da matriz Ybus: `G=real(Ybus)`; `B=imag(Ybus)`.
- Para implementação dos loops de resolução do problema, basta seguir o algoritmo disponibilizado nos slides.
- Como os subproblemas são definidos de acordo com o tipo de barra, é importante que se tenha apontadores para os números das barras associadas a cada tipo, bem como o número total de barras de cada tipo.

Exemplo:

SW – vetor de barras Swing → `SW=find(Tipo==1)`;

PV – vetor de barras PV → `PV=find(Tipo==2)`;

PQ – vetor de barras PQ → `PV=find(Tipo==3)`;

NSW – vetor de barras não SW → `NSW=find(Tipo~=1)`;

NPQ – vetor de barras não NPQ → `NPQ=find(Tipo~=3)`;

`nSW=length(SW)` - número da barra swings → `nSW=length(SW)`;

`nPV=length(PV)` - número de barras PV → `nPV=length(PV)`;

`nPQ=length(PQ)` - número de barras PQ → `nPQ=length(PQ)`;

`nNSW=length(NSW)` - número de barras não swing → `nNSW=length(NSW)`;

`nNPQ=length(NPQ)` - número de barras não PQ → `nNPQ=length(NPQ)`;

- A tolerância deve ser verificada para todas as variáveis $dP < tol$ e $dQ < tol$ associadas ao subproblema 1, além de verificar um número máximo de iterações para evitar loop infinito, caso o problema seja divergente, $it < N_{max}$.
- Após a solução do subproblema 1, resolva o subproblema 2.

Os resultados devem ser apresentados em duas tabelas: resultados de barra e resultados de circuitos. Os resultados de barra compreendem as tensões e ângulos. Os resultados de circuitos compreendem os fluxos de potência ativa e reativa. Imprima os resultados considerando potências em MW e MVar; tensões em pu e ângulos em graus.