TP N° :3

Application de la méthode de Gaus Siedel pour l'écoulement de puissance.

OBJECTIF :

Comprendre, la formulation mathématique du modèle d'écoulement de puissance dans sa forme complexe, et utiliser une méthode simple pour résoudre le problème de flux de puissance d'un système de petite taille en utilisant l'algorithme itératif de Gauss-Seidel.

LOGICIEL : MATLAB 7

THEORIE :

La méthode GAUSS - SEIDEL est un algorithme itératif pour résoudre le système d'équations non linéaires de flux de puissances.

Le système d'équations est donné par :

$$V_i^{r+1} = \frac{1}{Y_{ii}}\left[\frac{P_i - jQ_i}{\hat{V}_i^r} - \sum_{k=1}^{i-1} Y_{ik}V_k^{r+1} - \sum_{k=i+1}^{NP} Y_{ik}V_k^r\right]$$

La puissance réactive de jeu de barres de génération.

$$Q_i = -Im\left\{V_i^* \sum_{k=1}^{N} Y_{ik}V_k\right\}$$

Préparation de données :

Données des lignes

Linedata=[bus départ , bus arrivé , Résistance, Réactance, ½ susceptance, code ligne (1 pour ligne ou rapport du transformateur régulateur]

Données des jeux de barres

Busdata=[N° , code, module tension, angle, Pcharge , Qcharge, Pgénérateur, Q_{min} générateur, Q_{max}générateur, Q injectée]

Code Programme :

```
clc
clear all
basemva=100;
linedata=[1  2   0.01008    0.0504    0.05125    1
          1  3   0.00744      0.0372    0.03875 1
          2  4  0.00744      0.0372    0.03875   1
          3  4  0.01272     0.0636    0.06375    1]

 j=sqrt(-1); i = sqrt(-1);
nl = linedata(:,1); nr = linedata(:,2); R = linedata(:,3);
X = linedata(:,4); Bc = j*linedata(:,5); a = linedata(:, 6);
nbr=length(linedata(:,1)); nbus = max(max(nl), max(nr));
Z = R + j*X;
y= ones(nbr,1)./Z;  % admittance de braches
for n = 1:nbr
if a(n) <= 0  a(n) = 1; else end
Ybus=zeros(nbus,nbus);     % initialization de  Ybus à zero
               % formation des elements hors la diagonale
for k=1:nbr;
       Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k)/a(k);
       Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
    end
end
              % formation des elements de la diagonale
for  n=1:nbus
     for k=1:nbr
         if nl(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k)/(a(k)^2) + Bc(k);
         elseif nr(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k) +Bc(k);
         else, end
     end
end
  busdata=[1  0  1   0   50  30.99  0   0    0   0   0
         2  1  1   0    170 105.35  0   0    0   0   0
         3  1   1   0   200   123.94   0   0   0   0   0
         4  2   1.02  0  80  49.58  318 0  0   0   0  ]
Vm=0; delta=0; yload=0; deltad =0;
nbus = length(busdata(:,1));
for k=1:nbus
n=busdata(k,1);
kb(n)=busdata(k,2); Vm(n)=busdata(k,3); delta(n)=busdata(k, 4);
Pd(n)=busdata(k,5); Qd(n)=busdata(k,6); Pg(n)=busdata(k,7); Qg(n) =
busdata(k,8);
Qmin(n)=busdata(k, 9); Qmax(n)=busdata(k, 10);
Qsh(n)=busdata(k, 11);
    if Vm(n) <= 0  Vm(n) = 1.0; V(n) = 1 + j*0;
    else delta(n) = pi/180*delta(n);
```

```
        V(n) = Vm(n)*(cos(delta(n)) + j*sin(delta(n)));
        P(n)=(Pg(n)-Pd(n))/basemva;
        Q(n)=(Qg(n)-Qd(n)+ Qsh(n))/basemva;
        S(n) = P(n) + j*Q(n);
    end
DV(n)=0;
end
num = 0; AcurBus = 0; converge = 1;
Vc = zeros(nbus,1)+j*zeros(nbus,1); Sc = zeros(nbus,1)+j*zeros(nbus,1);

while exist('accel')~=1
   accel = 1.3;
end
while exist('accuracy')~=1
   accuracy = 0.001;
end
while exist('basemva')~=1
   basemva= 100;
end
while exist('maxiter')~=1
   maxiter = 100;
end
iter=0;
maxerror=10;
while maxerror >= accuracy & iter <= maxiter
iter=iter+1;
  for n = 1:nbus;
  YV = 0+j*0;
    for L = 1:nbr;
            if nl(L) == n, k=nr(L);
            YV = YV + Ybus(n,k)*V(k);
            elseif nr(L) == n, k=nl(L);
            YV = YV + Ybus(n,k)*V(k);
            end
    end
       Sc = conj(V(n))*(Ybus(n,n)*V(n) + YV) ;
       Sc = conj(Sc);
       DP(n) = P(n) - real(Sc);
       DQ(n) = Q(n) - imag(Sc);
         if kb(n) == 1
         S(n) =Sc; P(n) = real(Sc); Q(n) = imag(Sc); DP(n) =0; DQ(n)=0;
         Vc(n) = V(n);
         elseif kb(n) == 2
         Q(n) = imag(Sc); S(n) = P(n) + j*Q(n);

         if Qmax(n) ~= 0
             Qgc = Q(n)*basemva + Qd(n) - Qsh(n);
           if abs(DQ(n)) <= .005 & iter >= 10 % Après 10 iterations
             if DV(n) <= 0.045       % Les Mvardes bus generateur sont tester
                  if Qgc < Qmin(n),   % Si Vm(n) ne sont pas dans les limites
                  Vm(n) = Vm(n) + 0.005;  % sera changer d'un pas 0.005 pu
                  DV(n) = DV(n)+.005;     % jusqu'a .05  pu pour maintenir
                  elseif Qgc > Qmax(n),    % les Mvar des générateurs  dans
                  Vm(n) = Vm(n) - 0.005;    % les limites specifiees.
                  DV(n)=DV(n)+.005; end
               else, end
```

```
            else,end
          else,end
        end
      if kb(n) ~= 1
      Vc(n) = (conj(S(n))/conj(V(n)) - YV )/ Ybus(n,n);
      else, end
         if kb(n) == 0
         V(n) = V(n) + accel*(Vc(n)-V(n));
         elseif kb(n) == 2
         VcI = imag(Vc(n));
         VcR = sqrt(Vm(n)^2 - VcI^2);
         Vc(n) = VcR + j*VcI;
          V(n) = V(n) + accel*(Vc(n) -V(n));
         end
   end
  maxerror=max( max(abs(real(DP))), max(abs(imag(DQ))) );
   if iter == maxiter & maxerror > accuracy
   fprintf('\nWARNING: le prcessus iterative ne converge pas après ')
   fprintf('%g', iter), fprintf(' iterations.\n\n')
   fprintf('Appuiez Enter pour terminer lese iterations et imprimer les
résultats \n')
   converge = 0; pause, else, end

end
if converge ~= 1
   tech= ('               LA SOLUTION ITERATIVE NE CONVERGE PAS'); else,
   tech=('                  Power Flow Solution by Gauss-Seidel Method');
end
k=0;
for n = 1:nbus
  Vm(n) = abs(V(n)); deltad(n) = angle(V(n))*180/pi;
     if kb(n) == 1
     S(n)=P(n)+j*Q(n);
     Pg(n) = P(n)*basemva + Pd(n);
     Qg(n) = Q(n)*basemva + Qd(n) - Qsh(n);
     k=k+1;
     Pgg(k)=Pg(n);
     elseif  kb(n) ==2
     k=k+1;
     Pgg(k)=Pg(n);
     S(n)=P(n)+j*Q(n);
     Qg(n) = Q(n)*basemva + Qd(n) - Qsh(n);
     end
yload(n) = (Pd(n)- j*Qd(n)+j*Qsh(n))/(basemva*Vm(n)^2);
end
Pgt = sum(Pg);  Qgt = sum(Qg); Pdt = sum(Pd); Qdt = sum(Qd); Qsht = sum(Qsh);
busdata(:,3)=Vm'; busdata(:,4)=deltad';
clear  AcurBus  DP  DQ  DV  L Sc Vc VcI VcR YV converge delta
%affichage résulta
disp(tech)
fprintf('                     Maximum Power Mismatch = %g \n', maxerror)
fprintf('                            No. DES Iterations = %g \n\n', iter)
head =['    Bus  Tension  Angle    ----charge------    ---Generation---
Injected'
       '    No.  Mag.     Degree     MW       Mvar       MW       Mvar
Mvar '
```

```
       '
'];
disp(head)
for n=1:nbus
     fprintf(' %5g', n), fprintf(' %7.3f', Vm(n)),
     fprintf(' %8.3f', deltad(n)), fprintf(' %9.3f', Pd(n)),
     fprintf(' %9.3f', Qd(n)),  fprintf(' %9.3f', Pg(n)),
     fprintf(' %9.3f ', Qg(n)), fprintf(' %8.3f\n', Qsh(n))
end
    fprintf('      \n'), fprintf('    Total              ')
    fprintf(' %9.3f', Pdt), fprintf(' %9.3f', Qdt),
    fprintf(' %9.3f', Pgt), fprintf(' %9.3f', Qgt), fprintf(' %9.3f\n\n',
Qsht)
```