DEPARTMENT OF ELECTRICAL ENGINEERING

POWER SYSTEM OPERATION AND CONTROL LABORATORY MANUAL

Semester/Academic Year: VII Sem / 2017-18

Prepared By:

Rokhaiya Banu (Asst Prof,EE)

Under the Guidance of Program Owner

INDEX

EXPERIMENT NO:1

Y Bus formation by singular matrix transformation method using MATLAB programming

AIM: To develop a program to obtain Y_{bus} matrix for the given network by singular transformation method

SOFTWARE REQUIRED: MATLAB

THEORY:

Bus admittance is often used in power system studies. In most of the power system studies it is required to form Y- bus matrix of the system by considering certain power system parameters depending upon the type of analysis.

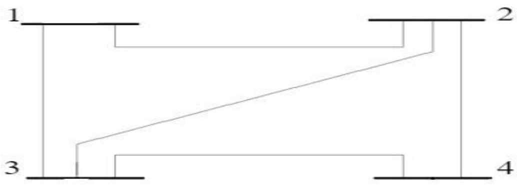
In power engineering, nodal admittance matrix (or just admittance matrix) or Y Matrix or Ybus is an *N x N* matrix describing a power system with *N* buses. It represents the nodal admittance of the buses in a power system. In realistic systems which contain thousands of buses, the Y matrix is quite sparse. Each bus in a real power system is usually connected to only a few other buses through the transmission lines. The Y Matrix is also one of the data requirements needed to formulate a power flow study.

PROCEDURE:

1. Enter the command window of the MATLAB.
2. Create a new M – file by selecting File - New – M – File
3. Type and save the program in the editor window.
4. Execute the program by either pressing Tools – Run.
5. View the results.

Line Data:

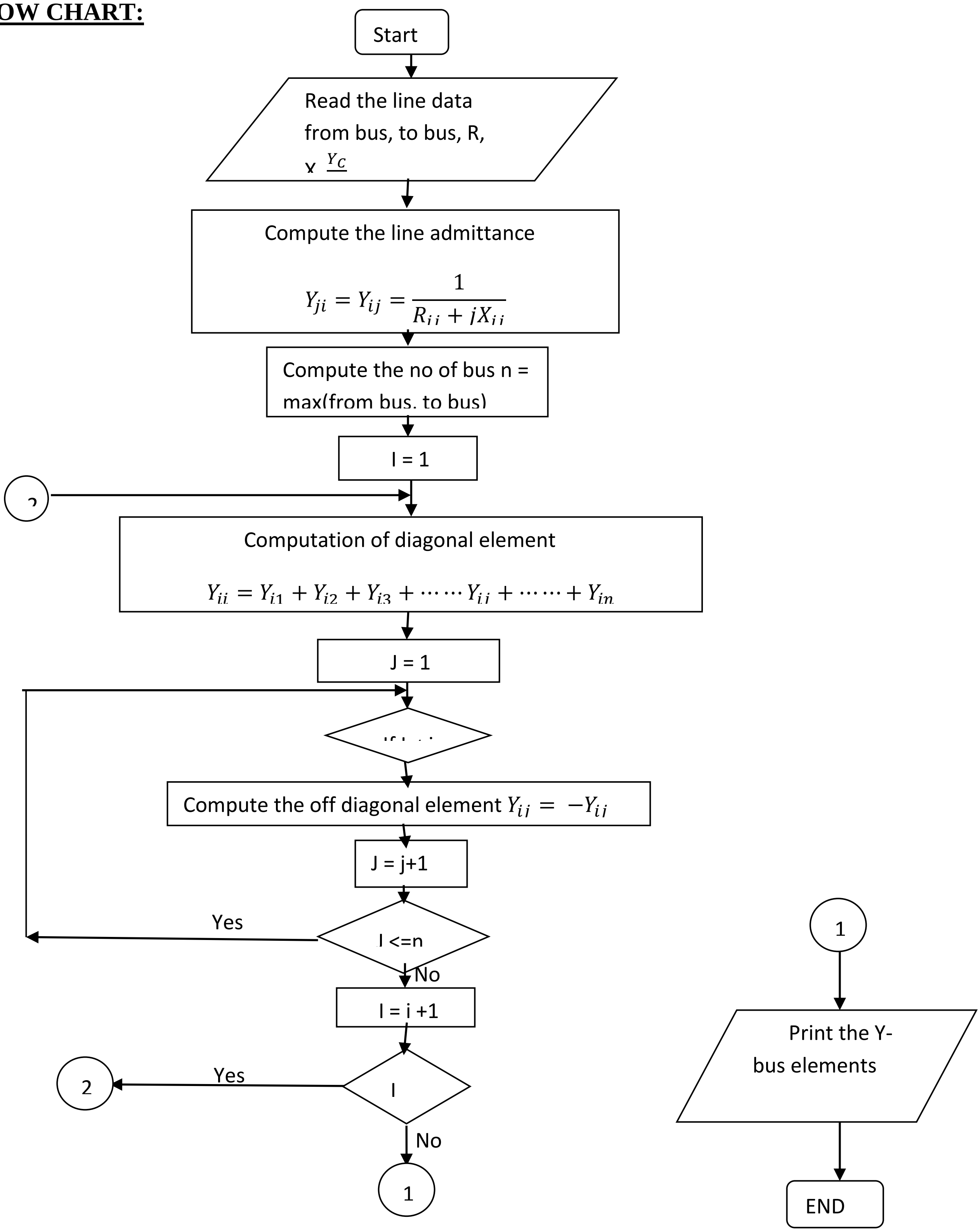
Sr no	Connected Bus		R	X
	SB	EB		
1	1	2	0.05	j0.15
2	1	3	0.1	j0.3
3	2	3	0.15	j0.45
4	2	4	0.10	j0.30
5	3	4	0.05	j0.15



PROGRAM:

```
clear all;
clc;
fp=fopen('y.out','w');
nb=4;
nl=5;
line=zeros(nl,1);
A=zeros(nl,nb);
prY=zeros(nl,nl);
linedata=[1 1 2 0.05+1i*0.15
          2 1 3 0.10+1i*0.30
          3 2 3 0.15+1i*0.45
          4 2 4 0.10+1i*0.30
          5 3 4 0.05+1i*0.15];
      sb=linedata(:,2);
      eb=linedata(:,3);
      for i=1:1:nl
         for j=1:1:nb
            p=sb(i);
           q=eb(i);
            A(i,p)=1.0;
            A(i,q)=1.0;
         end
      end
      At=A';
      for i=1:1:nl
         prY(i,i)=1.0/linedata(i,4);
      end
      Y=At*prY*A;
      fprintf(fp,'\n elemnets of Ybus');
      for i=1:1:nb
         for j=1:1:nb
            fprintf(fp,'\n %2d %2d %8.4f',i,j,real(Y(i,j)),imag(Y(i,j)));
         end
      end
      fclose(fp);
```

FLOW CHART:


Start
Read the line data from bus, to bus, R, X, $\frac{Y_C}{2}$
Compute the line admittance
$Y_{ji} = Y_{ij} = \frac{1}{R_{ij} + jX_{ij}}$
Compute the no of bus n = max(from bus, to bus)
I = 1
2
Computation of diagonal element
$Y_{ii} = Y_{i1} + Y_{i2} + Y_{i3} + \cdots\cdots Y_{ij} + \cdots\cdots + Y_{in}$
J = 1
If I ≠ j
Compute the off diagonal element $Y_{ij} = -Y_{ij}$
J = j+1
Yes
J <=n
No
I = i +1
Yes
2
I
No
1
1
Print the Y-bus elements
END

OUTPUT:

```
A =1    -1     0     0

     1     0    -1     0

     0     1    -1     0

     0     1     0    -1

     0     0     1    -1

A'= 1     1     0     0     0

      -1     0     1     1     0

       0    -1    -1     0     1

       0     0     0    -1    -1

prY =  [2.0000 - 6.0000i        0                  0                  0                 0

                   0            1.0000 - 3.0000i      0                 0                 0

                   0                 0           0.6667 - 2.0000i        0                  0

                   0                 0                      0         1.0000 - 3.0000i       0

                   0                 0                      0                  0         2.0000 - 6.0000i]


   Y =   [3.0000 - 9.0000i   -2.0000 +6.0000i    -1.0000 +3.0000i        0

          -2.0000 + 6.0000i   3.6667 -11.0000i   - 0.6667+2.0000i   -1.0000+ 3.0000

          -1.0000 + 3.0000i   -0.6667 + 2.0000i    3.6667 -11.0000i   -2.0000 +6.0000i

                     0         -1.0000+3.0000i     -2.0000 +6.0000i   3.0000 - 9.0000i]
```

RESULT: Thus the program for Ybus by singular transformation method was executed and the output is verified by manual calculation

EXPERIMENT NO:2

Y Bus formation by inspection method using MATLAB programming

AIM: To develop a program to obtain Y_{bus} matrix for the given networks by inspection method using MATLAB programming

SOFTWARE REQUIRED: MATLAB

THEORY: Bus admittance is often used in power system studies. In most of the power system studies it is required to form Y- bus matrix of the system by considering certain power system parameters depending upon the type of analysis.

Y-bus may be formed by inspection method, only if there is no mutual coupling between the lines. Every transmission line should be represented by Π - equivalent. Shunt impedances are added to diagonal element corresponding to the buses at which these are connected. The off diagonal elements are unaffected. The equivalent circuit of Tap changing transformers is included while forming Y-bus matrix.

FORMATIONOF Y-BUSMATRIX:

$$\text{Generalised Y-bus} = \begin{bmatrix} Y_{11} & \dots\dots\dots. & Y_{1n} \\ \vdots & \vdots & \vdots \\ Y_{n1} & \dots\dots & Y_{nn} \end{bmatrix}$$

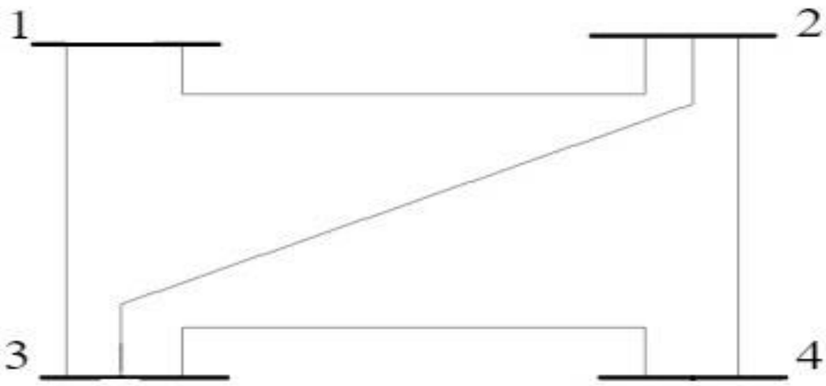
Each admittance Y_{ii} (i =1,2,……n) is called the self admittance or driving point admittance of bus I and equals the sum of all admittances terminating on the particular bus.
Each off-diagonal term Y_{ij} (i,j = 1,2…n; j□i) is the transfer admittance between buses I and j, n=total number of buses. Further, $Y_{ij} = Y_{ji}$

PROCEDURE:

1. Enter the command window of the MATLAB.
2. Create a new M – file by selecting File - New – M – File
3. Type and save the program in the editor window.
4. Execute the program by either pressing Tools – Run.
5. View the output.

LINEDATA:

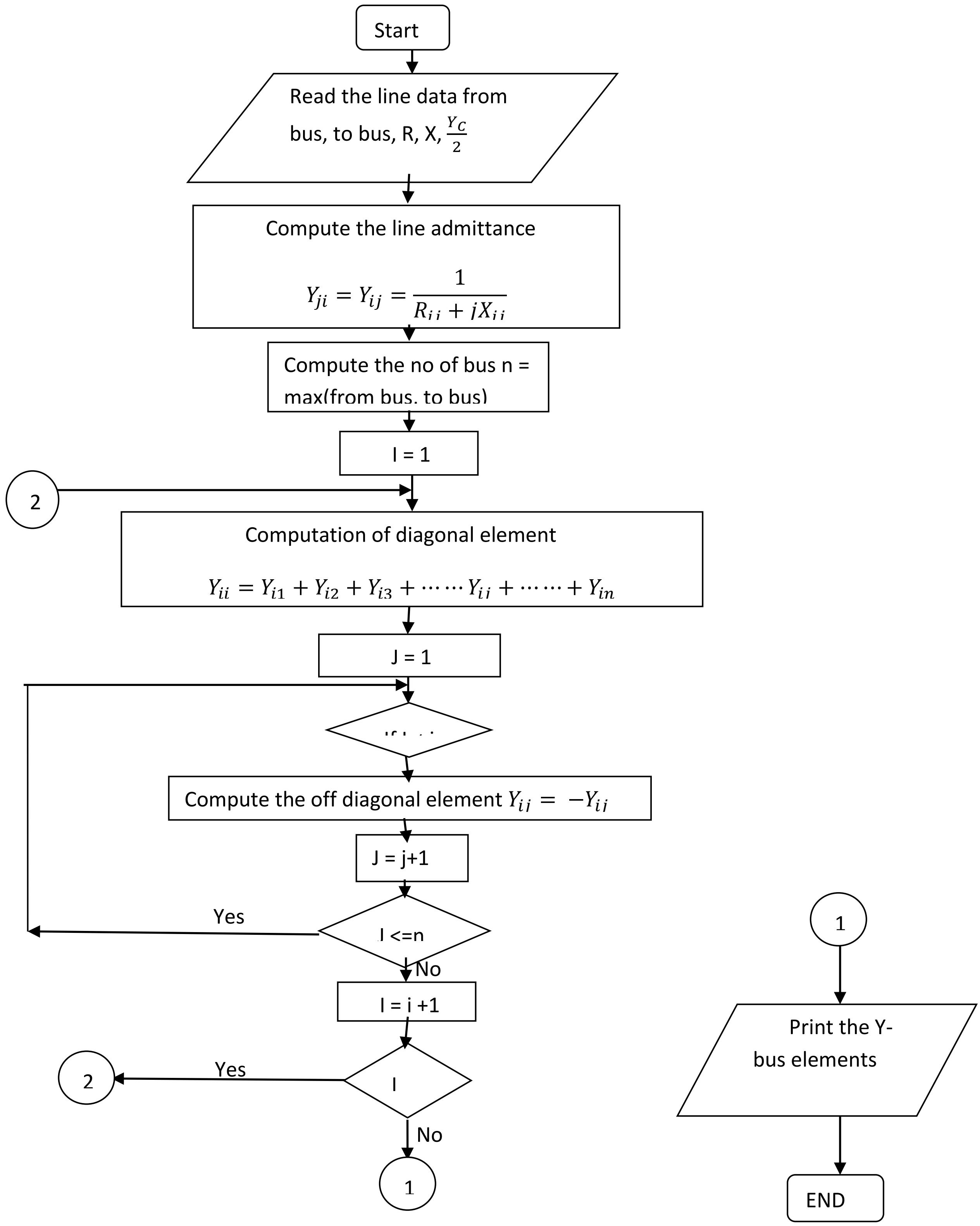
Line no	SB	EB	Impedance	½ LC	Tap ratio
1	1	2	0.05+j0.15	0.030j	1.0
2	1	3	0.10+j0.30	0.025j	1.0
3	2	3	0.15+j0.45	0	1.0
4	2	4	0.10+j0.30	0	1.0
5	3	4	0.15+j0.15	0	1.0



PROGRAM:

```
clc;
clear all;
fp=fopen('yb.out','w');
nl=5;
nb=4;
A=zeros(nl,1);
shty=zeros(nl,1);
sery=zeros(nl,1);
Y=zeros(nb,nb);
linedata=[1 1 2 0.05+1i*0.15 1i*0.030 1.0
          2 1 3 0.10+1i*0.30  1i*0.025 1.0
          3 2 3 0.15+1i*0.45      0      1.0
          4 2 4 0.10+1i*0.30      0      1.0
          5 3 4 0.15+1i*0.15      0      1.0];
    for i=1:nl
       A(i)=linedata(i,6);
       sery(i)=(1/(linedata(i,4)))/A(i);
       shty(i)=linedata(i,5);
       m=linedata(i,2);
       k=linedata(i,3);
       Y(m,m)=Y(m,m)+sery(i)+shty(i)+sery(i)*(1-A(i))/A(i);
       Y(k,k)=Y(k,k)+sery(i)+shty(i)+sery(i)*(A(i)-1);
       Y(m,k)=Y(m,k)-sery(i);
       Y(k,m)=Y(k,m)-sery(i);
    end
    fprintf(fp,'\nelements of Ybus');
    for i=1:1:nb
       for j=1:1:nb
          fprintf(fp,'\n %d %d %8.5f j%8.5f',i,j,real(Y(i,j)));
       end
    end
    fclose(fp);
```

FLOWCHART:


Start
Read the line data from bus, to bus, R, X, $\frac{Y_C}{2}$
Compute the line admittance
$Y_{ji} = Y_{ij} = \frac{1}{R_{ii} + jX_{ii}}$
Compute the no of bus n = max(from bus, to bus)
I = 1
2
Computation of diagonal element
$Y_{ii} = Y_{i1} + Y_{i2} + Y_{i3} + \cdots\cdots Y_{ii} + \cdots\cdots + Y_{in}$
J = 1
Compute the off diagonal element $Y_{ij} = -Y_{ij}$
J = j+1
Yes
I <=n
No
I = i +1
2
Yes
I
No
1
1
Print the Y-bus elements
END

OUTPUT:

Y= [3.0000 - 8.9450i   -2.0000 + 6.0000i   -1.0000 + 3.0000i   0

-2.0000 + 6.0000i   3.6667 -10.9700i   -0.6667 + 2.0000i   -1.0000 + 3.0000i

-1.0000 + 3.0000i   -0.6667 + 2.0000i   5.0000 - 8.3083i   -3.3333 + 3.3333i

0   -1.0000 + 3.0000i   -3.3333 + 3.3333i   4.3333 - 6.3333i]

RESULT: Thus the program for Ybus formation by inspection method is executed and the output is verified by manual calculation

EXPERIMENT NO:3

Load flow analysis by Gauss Seidal method using MATLAB programming

AIM: To carry load flow analysis of the given power system using Gauss Seidal method by MATLAB programming

SOFTWARE REQUIRED: MATLAB

THEORY: Load flow analysis is the study conducted to determine the steady state operating condition of the given system under given conditions. A large number of numerical algorithms have been developed and Gauss Seidel method is one of such algorithm.

The performance equation of the power system may be written of

[I bus] = [Y bus][V bus]

Selecting one of the buses as the reference bus, we get (n-1) simultaneous equations. The bus loading equations can be written as

$I_i = P_i - jQ_i / V_i^*$ (i=1,2,3,…………..n)

$$P_i = \mathrm{Re}\left[\sum_{k=1}^{n} V_i^* Y_{ik} V_k\right].$$

$$Q_i = -\mathrm{Im}\left[\sum_{k=1}^{n} V_i^* Y_{ik} V_k\right].$$

The bus voltage can be written in form of

$$V_i = (1.0/Y_{ii})\left[I_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{ij} V_j\right]$$

j≠i(i=1,2,…………n)& i≠slack bus

Substituting Ii in the expression for Vi, we get

$$V_{i\,new} = (1.0/Y_{ii})\left[P_i - JQ_i / V_{io}^* - \sum_{J=1}^{n} Y_{ij} V_{io}\right]$$

The latest available voltages are used in the above expression, we get

$$V_{i\,new} = (1.0/Y_{ii})\left[P_i - JQ_i / V_i^{o*} - \sum_{J=1}^{n} Y_{ij} V_j^{n} - \sum_{j=i+1}^{n} Y_{ij} V_{io}\right]$$

The above equation is the required formula .this equation can be solved for voltages interactive manner. During each iteration, we compute all the bus voltage and check for convergence is carried out by comparison with the voltages obtained at the end of previous iteration. After the solutions is obtained. The stack bus real and reactive powers, the reactive power generation at other generator buses and line flows can be calculated.

ALGORITHM:

Step1: Read the data such as line data, specified power, specified voltages, Q limits at the generator buses and tolerance for convergences

Step2: Compute Y-bus matrix.

Step3: Initialize all the bus voltages

Step4: Iteration=1

Step5: Consider i=2, where i' is the bus number

Step6: check whether this is PV bus or PQ bus. If it is PQ bus go to step 8 otherwise go to next step.

Step7: Compute Qi check for q limit violation. QGi=Qi+QLi.

a).If QGi>Qi max ,equate QGi = Qimax. Then convert it into PQ bus.

b).If QGi<Qi min, equate QGi = Qi min. Then convert it into PQ bus

Step8: Calculate the new value of the bus voltage using gauss seidal formula.

$$V_i=(1.0/Y_{ii})\left[(P_i-j\,Q_i)/v_i^{0*}-\sum_{J=1}^{i=1} Y_{ij}V_j-\sum_{J=i+1}^{n} Y_{ij}V_j^{0}\right]$$

Adjust voltage magnitude of the bus to specify magnitude if Q limits are not violated

Step9: If all buses are considered go to step 10 otherwise increments the bus no. i=i+1 and Go to step6.

Step10: Check for convergence. If there is no convergence goes to step 11 otherwise go to step12

Step11: Update the bus voltage using the formula.

$Vi_{new}=Vi_{old}+\alpha(vi_{new}-Vi_{old})$ (i=1,2,…..n) i≠ slackbus ,α is the acceleration factor=1.4

Step12: Calculate the slack bus power, Q at P-V buses real and reactive give flows real and reactance line losses and print all the results including all the bus voltages and all the bus angles

Step 13: Stop

PROCEDURE:

1. Enter the command window of the MATLAB.
2. Create a new M – file by selecting File - New – M – File
3. Type and save the program in the editor window.
4. Execute the program by either pressing Tools – Run.
5. View the output.

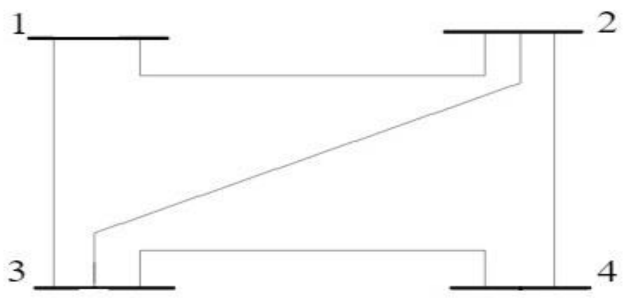
LINEDATA:

Ybus

3-9j	-2-6i	-1+3j	0
-2-6j	3.66-11j	-0.666+2i	-1+3j
-1+3j	-0.666+2j	3.66-11j	-2+6j
0	-1+3j	-2+6j	3-9j

Bus Data

Bus no	P	Q	|V|	Bus Type
1	-	-	1.04	Slack
2	0.5	-0.2	-	PQ
3	-1.0	0.5	-	PQ
4	0.3	-0.1	-	PQ



PROGRAM:

```
clc;
clear all;
nb=4;
V=[1.04 1 1 1];
Y=[3-1i*9    -2+1i*6       -1+1i*3       0
  -2+1i*6  3.66-1i*11  -0.666+1i*2 -1+1i*3
  -1+1i*3  -0.666+1i*2  3.66-1i*11 -2+1i*6
     0    -1+1i*3        -2+1i*6     3-1i*9];
P=[inf 0.5 -1 0.3];
Q=[inf -0.2 0.5 -0.1];
for n=1:3
    for i=2:nb
        temp=0;
        for k=1:nb
            if i~=k
                temp=temp+(Y(i,k)*V(k));
            end
        end
        S=(P(i)-1i*Q(i))/conj(V(i));
        V(i)=(S-temp)/Y(i,i);
    end
    n;
    V;
end
```

OUTPUT:

V= [1.0400   1.0395+0.0166i   1.0468-0.1063i   1.0454-0.0335i]

RESULT: The load flow analysis by Gauss Siedal method for the following power system is executed

EXPERIMENT NO:4

Load flow analysis by NR method using MATLAB programming

AIM: To carry load flow analysis of the given power system using NR method by MATLAB programming

SOFTWARE REQUIRED: MATLAB

THEORY:

The Newton Raphson method of load flow analysis is an iterative method which approximates the set of non-linear simultaneous equations to a set of linear simultaneous equations using Taylor's series expansion and the terms are limited to first order approximation. The load flow equations for Newton Raphson method are non-linear equations in terms of real and imaginary part of bus voltages.

ALGORITHM:

Step1: Input the total number of buses. Input the details of series line impendence and line charging admittance to calculate the Y-bus matrix.

Step2: Assume all bus voltage as 1 per unit except slack bus.

Step3: Set the iteration count as k=0 and bus count as p=1.

Step4: Calculate the real and reactive power pp and qp using the

formula P=ΣvpqYpq*cos(Qpq+εp-εq)

Qp=ΣVpqYpa*sin(qpq+εp-εa)

Step5: If the bus is generator (PV) bus, check the value of Qp*is within the limits.If it Violates the limits, then equate the violated limit as reactive power and treat it as PQ bus. If limit is not isolated then calculate,

|vp|^r=|vgp|^rspe-|vp|r ;   Qp*=qsp-qp*

Step6: Advance bus count by 1 and check if all the buses have been accounted if not go to step5.

Step7: Calculate the elements of Jacobean matrix.

Step8: Calculate new bus voltage increment pk

and fpk

Step9: Calculate new bus voltage ep*h+ ep*

Fp^k+1=fpK+fpK

Step10: Advance iteration count by 1 and go to step3.

Step11: Evaluate bus voltage and power flows through the line

PROCEDURE:

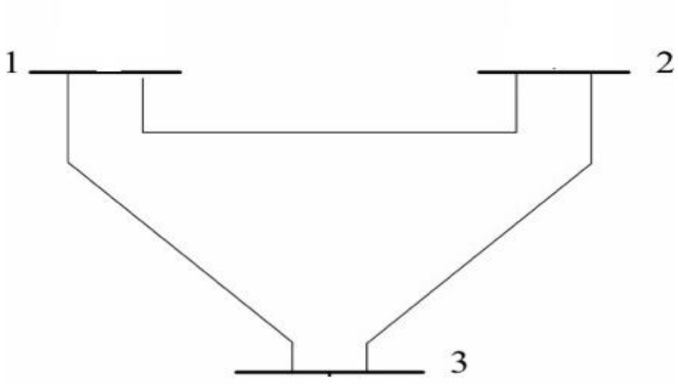
1. Enter the command window of the MATLAB.
2. Create a new M – file by selecting File - New – M – File
3. Type and save the program in the editor window.
4. Execute the program by either pressing Tools – Run.
5. View the output.

LINEDATA:

YBus

5.882-j23.51	-2941+j11.76	-2941+j11.76
-2941+j11.76	5.882-j23.51	-2941+j11.76
-2941+j11.76	-2941+j11.76	5.882-j23.51

Bus No	P	Q	|V|	delta	Bus type
1	-0.23	-0.96	1.0	0	PQ
2	-0.12	-0.16	1.0	0	PQ
3	-	-		-	Slack



PROGRAM:

```
clc;
clear all;
data=[1 -0.23 -0.96 1.0 0
      2 -0.12 -0.16 1.0 0];
  Y=[5.882-1i*23.51 -2.941+1i*11.76 -2.941+1i*11.76
    -2.941+1i*11.76 5.882-1i*23.51  -2.941+1i*11.76
    -2.941+1i*11.76 -2.941+1i*11.76 5.882-1i*23.41];
P(:,1)=data(:,2);
Q(:,1)=data(:,3);
V(:,1)=data(:,4);
delta(:,1)=data(:,5);
G(:,:)=real(Y(:,:));
B(:,:)=imag(Y(:,:));
n=2;
H=zeros(n,n);
N=zeros(n,n);
J=zeros(n,n);
L=zeros(n,n);
for i=1:n
    for j=1:n
        if i==j
            H(i,i)= -Q(i)-B(i,i)*V(i)^2;
            N(i,i)= P(i)+G(i,i)*V(i)^2;
            J(i,i)=P(i)-G(i,i)*V(i)^2;
            L(i,i)=Q(i)-B(i,i)*V(i)^2;
        else
            del=delta(i)-delta(j);
            H(i,j)=V(i)*V(j)*(G(i,j)*sin(del)-B(i,j)*cos(del));
            L(i,j)=H(i,j);
            N(i,j)=V(i)*V(j)*(G(i,j)*cos(del)+B(i,j)*sin(del));
        end
    end
end
Ja=[H N;J L];
```

OUTPUT:

H = 24.4700   -11.7600

  -11.7600   23.6700

N = 5.6520   -2.9410

  -2.9410   5.7620

J =   -6.1120   0

      0   -6.0020

L = 22.5500 -11.7600

-11.7600 23.3500

Ja = [24.4700 -11.7600 5.6520 -2.9410

-11.7600 23.6700 -2.9410 5.7620

-6.1120 0 22.5500 -11.7600

0 -6.0020 -11.7600 23.3500]

RESULT: The load flow analysis by NR method for the following power system is executed

EXPERIMENT NO:5

Optimum loading of generators neglecting transmission losses using MATLAB programming

AIM: To find optimum loading of generators neglecting transmission losses for two units using MATLAB programming

SOFTWARE REQUIRED: MATLAB

THEORY:

Economic Dispatch is the process of determination of the output power generated by the unit or units to supply the specified load in a manner that will minimize the total cost of fuel. Each generating unit has a unique production cost defined by its fuel cost coefficients. Economic Dispatch models the electric power system (with one or more control areas) and dispatches the available generation resources to supply a given load for each control area in the most economic manner in real-time operation. The objective is to minimize the total generation cost (including fuel cost, plus emission cost, plus operation/maintenance cost, plus network loss cost) by meeting the operational constraints

PROBLEM:

Incremental fuel costs in rupees per MWh for a plant consisting of two units are:

dP_1
dF_1 = 0.2P1+40

dP_2
dF_2 = 0.25P2+30

Assume that both units are operating at all times, and total load varies from 40 MW to 250 MW, and the maximum and minimum loads on each unit are to be 125 MW and 20 MW respectively. How will the load be shared between the two units as the system load varies over the full range? What are the corresponding values of the plant incremental costs?

PROCEDURE:

1. Enter the command window of the MATLAB.
2. Create a new M – file by selecting File - New – M – File
3. Type and save the program in the editor window.
4. Execute the program by either pressing Tools – Run.
5. View the output

PROGRAM:

```
clc;
clear all;
n=2;
Pd=231.25;
alpha=[0.20 0.25];
beta=[40 30];
lamda=20;
lamdaprev=lamda;
eps=1;
deltalamda=0.25;
Pgmax=[125 125];
Pgmin=[20 20];
Pg=100*ones(n,1);
while abs(sum(Pg)-Pd)>eps
    for i=1:n
        Pg(i)=(lamda-beta(i))/alpha(i);
        if Pg(i)>Pgmax(i)
            Pg(i)=Pgmax(i);
        end
        if Pg(i)<Pgmin(i);
            Pg(i)=Pgmin(i);
        end
    end
    if(sum(Pg)-Pd)<0
        lamdaprev=lamda;
        lamda=lamda+deltalamda;
    else
        lamdaprev=lamda;
        lamda=lamda-deltalamda;
    end
end
lamdaprev;
Pg;
```

OUTPUT:

lamdaprev = 61.2500

Pg =106.2500

125.0000

RESULT: Thus the optimum loading of generators neglecting losses is determined by executing the MATLAB program and verified output by manual calculation

EXPERIMENT NO: 6

Simulation of transient stability on single machine of power system

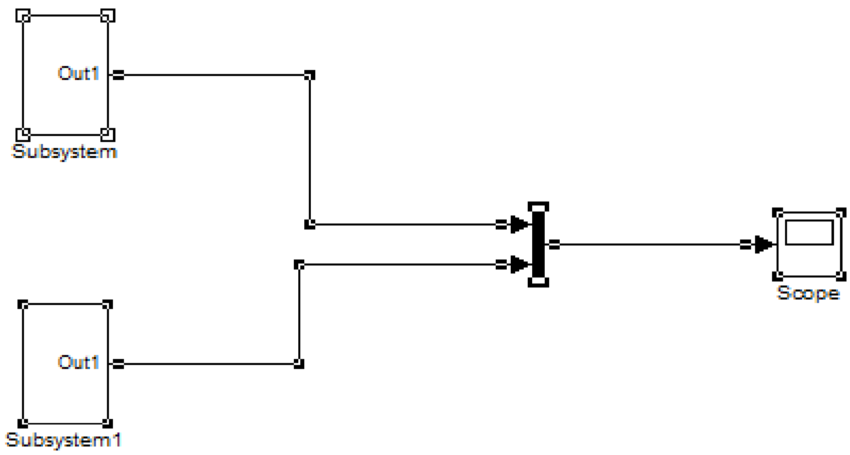
AIM: To model simulation of transient stability on single machine of power system using MATLAB

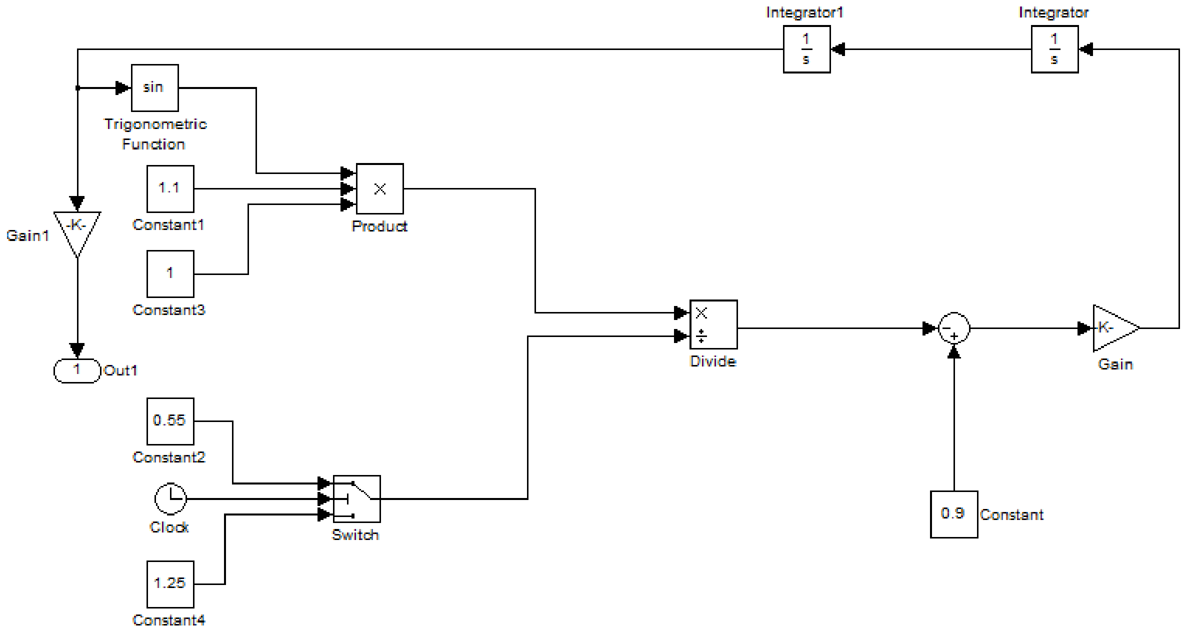
SOFTWARE REQUIRED: MATLAB

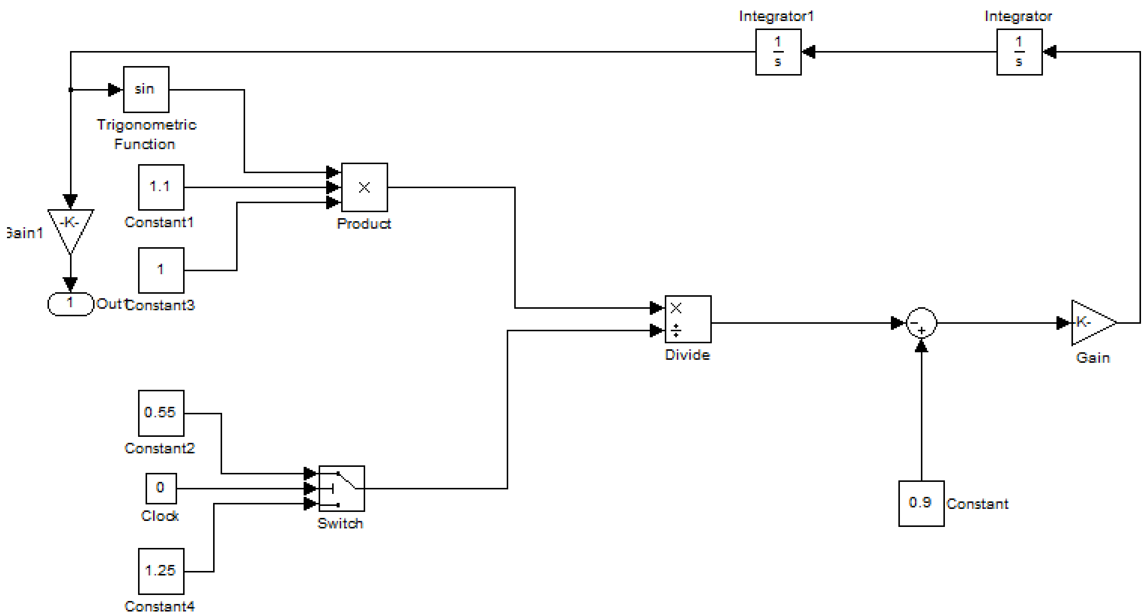
THEORY: Power system stability is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most of the system variables bounded so that practically the entire system remain intact

Transient stability in power system are done over a very small period of time equal to the time required for one swing, which approximates to around 1 sec or even less. If the system is found to be stable during this first swing, it's assumed that the disturbance will reduce in the subsequent swings, and the system will be stable thereafter as is generally the case.

SIMULINK MODEL:




Integrator1
Integrator
sin
Trigonometric Function
1.1
Constant1
1
Constant3
Product
Gain1
-K-
Out1
Divide
Gain
0.55
Constant2
Clock
Switch
1.25
Constant4
0.9
Constant


Integrator1
Integrator
sin
Trigonometric Function
1.1
Constant1
Product
1
Constant3
Out1
Divide
Gain
0.55
Constant2
0
Clock
Switch
1.25
Constant4
0.9
Constant

EXPECTED OUTPUT: