Computer Measurements in Power Systems (ET – 409)

LIST OF EXPERIMENTS

PROGRAM:-1

(A) Write a program to calculate Geometric mean radius (GMR) and geometric mean distance (GMD) of the following:-

(a) Single phase lines

AIM:-

To find GMD and GMR using matlab code.

PROGRAM:-

```
% SINGLE PHASE LINE
clc;
clear all;
D=input('distance between two conductors = ');
r=input('radius of conductor = ');
GMD=D;GMR=0.7788*r;
GMR
GMD
```

RESULT:-

```
distance between two conductors = 10
radius of conductor = 2

GMD =

    10


GMR =

    1.5576
```

(b) Three phase lines

1- Symmetrical spacing

AIM:-

To find GMD and GMR using matlab code.

PROGRAM:-

```
% THREE PHASE SYMMETRICAL LINE
clc;
clear all;
D=input('distance between any two conductors = ');
r=input('radius of conductor = ');
E0=8.854187817*10^-12;
GMD=D;GMR=0.7788*r;
GMR
GMD
```

RESULT:-

```
distance between two conductors = 45
radius of conductor = 3

GMR =

    2.3364


GMD =

    45
```

2- Asymmetrical spacing

AIM:-

To find GMD and GMR using matlab code.

PROGRAM:-

```
% ASYMETRECAL SPACING
clc;
clear all;
r=input('radius of conductor = ');;
Dab=6;Dac=12;Dbc=6;Dba=Dab;Dca=Dac;Dcb=Dbc;
GMD=(Dab*Dac*Dbc*Dba*Dca*Dcb)^(1/6);
GMR=0.7788*r;
GMR
GMD
```

RESULT:-

```
radius of conductor = 1.8

GMR =

    1.4018


GMD =

    7.5595
```

3- Bundled conductors

<u>AIM:-</u>

To find GMD and GMR using matlab code.

<u>PROGRAM:-</u>

```
% BUNDLED CONDUCTOR
clc;
clear all;
Daa=0.016*0.7788;
Da1a1=Daa;
Daa1=0.45;Da1a=0.45;
GMR=(Daa*Daa1*Da1a1*Da1a)^(1/4);
dab=12;dab1=12.45;da1b=11.55;da1b1=12;
Dab=(dab*dab1*da1b*da1b1)^(1/4);
Dbc=Dab;
dca=24;dca1=23.55;dc1a=24.45;dc1a1=24;
Dca=(dca*dca1*dc1a*dc1a1)^(1/4);
GMD=(Dab*Dbc*Dca)^(1/3);
GMR
GMD
```

<u>RESULT:-</u>

```
GMR =

    0.0749


GMD =

   15.1151
```

4- Three phase double circuit lines

AIM:-

To find GMD and GMR using matlab code.

PROGRAM:-

```
% DOUBLE CIRCUIT VERTICAL CONFIGURATION
clc;
clear all;
E0=8.854187817*10^-12;
r=input('enter radius of conductor = ');
h=input('enter the distance between b and b1 = ');
m=input('enter distance between a and b1 = ');
n=input('enter distance between a and a1 or c and c1 = ');
D=input('enter the distance between a and b1 = ');
r1=0.7788*r;
Dsa=sqrt(r1*n);Dsb=sqrt(r1*h);Dsc=Dsa;
GMR=(Dsa*Dsb*Dsc)^(1/3);
Dab=sqrt(D*m);
Dbc=Dab;Dca=sqrt(2*D*h);
GMD=(Dab*Dbc*Dca)^(1/3);
GMR
GMD
```

RESULT:-

```
enter radius of conductor = 2
enter the distance between b and b1 = 10
enter distance between a and b1 = 20
enter distance between a and a1 or c and c1 = 30
enter the distance between a and b1 = 10

GMR =

    5.6920


GMD =

   14.1421
```

(B) Find inductance and capacitance using above programs

<u>1. Single phase lines:</u> find the loop inductance and reactance per km of a single phase overhead line, consisting of two conductors, each 1.213 cm diameter. The spacing between conductors is 1.25 meters and frequency is 50 Hz.

<u>AIM:-</u>

To find inductance and capacitance using matlab code.

<u>PROGRAM:-</u>

```
% SINGLE PHASE LINE
clc;
clear all;
D=input('distance between two conductors = ');
r=input('radius of conductor = ');
E0=8.854187817*10^-12;
GMD=D;GMR=0.7788*r;
L=4*(10^-7)*log(GMD/GMR);
disp('SINGLE PHASE LINE INDUCTANCE AND CAPACITANCE');
L
C=2*pi*E0/(log(GMD/GMR));

C
```

<u>RESULT:-</u>

```
distance between two conductors = 56

radius of conductor = 1.7

SINGLE PHASE LINE INDUCTANCE AND CAPACITANCE

L = 1.4979e-06

C = 1.4856e-11
```

2. Three phase symmetrical: A three phase transmission line has its conductors at the corners of an equilateral triangle with side 3m. the diameter of each conductor is 1.63 cm. find the inductance per phase per kilometer of the line.

AIM:-

To find inductance and capacitance using matlab code.

PROGRAM:-

```
% THREE PHASE LINE
clc;
clear all;
D=input('distance between any two conductors = ');
r=input('radius of conductor = ');
E0=8.854187817*10^-12;
GMD=D;GMR=0.7788*r;
L=2*(10^-7)*log(GMD/GMR)*(10^3);     % H/Km per phase
disp('THREE PHASE LINE INDUCTANCE AND CAPACITANCE');
L
C=2*pi*E0/(log(GMD/GMR));
C
```

RESULT:-

```
distance between any two conductors = 34

radius of conductor = 1.6

THREE PHASE LINE INDUCTANCE AND CAPACITANCE

L =   6.6127e-04

C = 1.6826e-11
```

3. Asymmetrical spacing: A three phase transmission line has a horizontal configuration with a spacing of 6m between adjacent conductors and 12 m between outer conductors. The radius of each conductor is 1.81 cm. find the inductance per phase per km of the line.

AIM:-

To find inductance capacitance using matlab code.

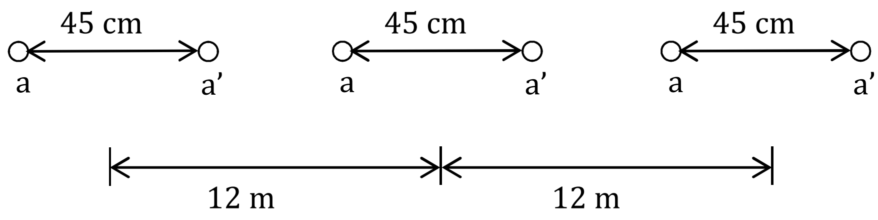
PROGRAM:-

```
% ASYMETRECAL SPACING
clc;
clear all;
r=1.81*10^-2;
E0=8.854187817*10^-12;
Dab=6;Dac=12;Dbc=6;Dba=Dab;Dca=Dac;Dcb=Dbc;
GMD=(Dab*Dac*Dbc*Dba*Dca*Dcb)^(1/6);
GMR=0.7788*r;
L=2*(10^-7)*log(GMD/GMR)*(10^3);     % H/Km per phase
disp('ASYMETRICAL SPACING INDUCTANCE AND CAPACITANCE');
L
C=2*pi*E0/(log(GMD/GMR));
C
```

RESULT:-

```
ASYMETRICAL SPACING INDUCTANCE AND CAPACITANCE

L =    0.0013

C =   8.8521e-12
```

3. Bundled conductor: A 400 kV 3-phase bundled conductor line with two sub-conductors per phase has a horizontal configuration as shown in fig.1.1. the radius of each sub-conductor is 1.6cm. (a) find the inductance per phase per km of the line. (b) compute the inductance of the line with only one conductor per phase having the same cross-sectional area of the conductor of each phase.



Three phase line with bundled conductors

AIM:-

To find inductance capacitance using matlab code.

PROGRAM:-

```
% BUNDLED CONDUCTOR
clc;
clear all;
E0=8.854187817*10^-12;
Daa=0.016*0.7788;
Da1a1=Daa;
Daa1=0.45;Da1a=0.45;
GMR=(Daa*Daa1*Da1a1*Da1a)^(1/4);
dab=12;dab1=12.45;da1b=11.55;da1b1=12;
Dab=(dab*dab1*da1b*da1b1)^(1/4);
Dbc=Dab;
dca=24;dca1=23.55;dc1a=24.45;dc1a1=24;
Dca=(dca*dca1*dc1a*dc1a1)^(1/4);
GMD=(Dab*Dbc*Dca)^(1/3);
L=2*(10^-7)*log(GMD/GMR)*(10^3);     % H/Km per phase
disp('BUNDLED CONDUCTOR INDUCTANCE AND CAPACITANCE');
L
C=2*pi*E0/(log(GMD/GMR));
C
r=sqrt(2)*0.016;r1=0.7788*r;
GMR=r1;
Dab=12;Dbc=12;Dca=24;Dba=12;Dcb=12;Dac=24;
GMD=(Dab*Dbc*Dca*Dba*Dcb*Dac)^(1/6);
L=2*(10^-7)*log(GMD/GMR)*(10^3);     % H/Km per phase
```

```
disp('INDUCTANCE AND CAPACITANCE WITH ONLY ONE CONDUCTOR PER
PHASE');
L
C=2*pi*E0/(log(GMD/GMR));
C
```

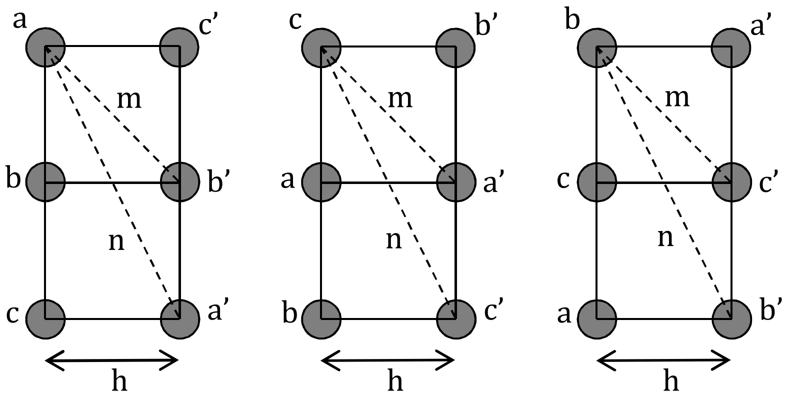
RESULT:-

```
BUNDLED CONDUCTOR INDUCTANCE AND CAPACITANCE

L =     0.0011

C =    1.0482e-11


INDUCTANCE AND CAPACITANCE WITH ONLY ONE CONDUCTOR PER PHASE

L =    0.0014

C =    8.2363e-12
```

4. Double circuit vertical configuration: fig 1.2 shows the three sections of a double circuit three-phase line with vertical spacing, over a transposition cycle. The conductors a and a' belong to one phase, b and b' to the second phase and c and c' to the third phase. In each case the conductors of two phases are placed diametrically opposite to each other and those of the third phase are placed horizontally opposite to each other. This configuration gives high value of GMR. To calculate inductance it is necessary to determine D_{eq} or GMD and D or GMR.

GMR of conductors of phase a in section 1 is

$$D_{sa} = \sqrt[4]{r'.n.r'.n} = \sqrt{r'.n}$$



Double circuit vertical configuration

AIM:-

To find inductance and capacitance using matlab code.

PROGRAM:-

```
% DOUBLE CIRCUIT VERTICAL CONFIGURATION
clc;
clear all;
E0=8.854187817*10^-12;
r=input('enter radius of conductor = ');
h=input('enter the distance between b and b1 = ');
m=input('enter distance between a and b1 = ');
n=input('enter distance between a and a1 or c and c1 = ');
D=input('enter the distance between a and b1 = ');
r1=0.7788*r;
Dsa=sqrt(r1*n);Dsb=sqrt(r1*h);Dsc=Dsa;
GMR=(Dsa*Dsb*Dsc)^(1/3);
Dab=sqrt(D*m);
```

```
Dbc=Dab;Dca=sqrt(2*D*h);
GMD=(Dab*Dbc*Dca)^(1/3);
L=2*(10^-7)*log(GMD/GMR)*(10^3);     % H/Km per phase
disp('INDUCTANCE AND CAPACITANCE OF DOUBLE CIRCUIT VERTICAL
CONFIGURATION');
L
C=2*pi*E0/(log(GMD/GMR));
C
```

RESULT:-

```
enter radius of conductor = 1.6

enter the distance between b and b1 = 12

enter distance between a and b1 = 12

enter distance between a and a1 or c and c1 = 12

enter the distance between a and b1 = 12

INDUCTANCE AND CAPACITANCE OF DOUBLE CIRCUIT VERTICAL
CONFIGURATION

L =  2.4960e-04

C =  4.4578e-11
```

PROGRAM – 2

Q) Write a general computer program for n number of buses and elements to develop

- **a Bus incidence matrix (A)**
- **b Basic cut-set incidence matrix (B)**
- **c Basic loop incidence matrix (C)**

AIM:-

Formation of network matrices A, B, C.

THEORY:-

Incidence matrix (A)

The incidence of branches to nodes in a connected graph is given by the element-node incidence matrix, A'. An element a_{ij} of A' is defined as under:

= 1; if the element i is oriented away from the node j.
= -1; if the element i is oriented towards the node j.
= 0; if the element i is not at all incident on the node j.

Cut-set matrix (B)

The incidence of elements to basic cut-sets of a connected graph is shown by the basic cut-set matrix B. The elements of this matrix are:

= 1; if the i^{th} element is incident to and oriented in the same direction as the j^{th} basic cutset
= -1; if the i^{th} element is incident to and oriented in the opposite direction as the j^{th} basic cutset
= 0; if the i^{th} element is not incident to the j^{th} basic cutset

Loop matrix (C)

The incidence of elements to basic loops of a connected graph is shown by the basic loop incidence matrix C. The elements of this matrix are:

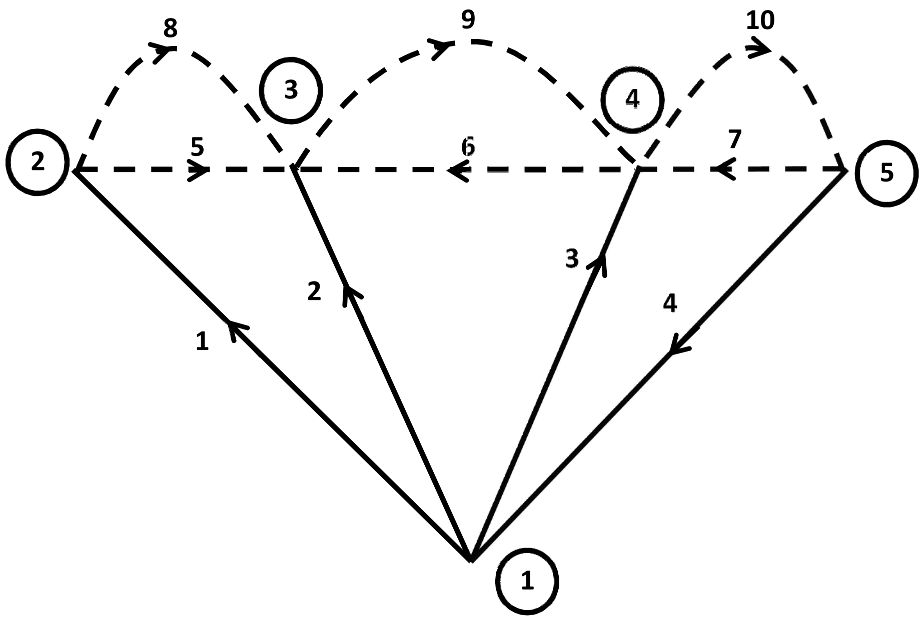
= 1; if the i^{th} element is incident to and oriented in the same direction as the j^{th} basic loop
= -1; if the i^{th} element is incident to and oriented in the opposite direction as the j^{th} basic loop

= 0; if the i^{th} element is not incident to the j^{th} basic loop

$B_l=A_l*(A_b)^{-1}$

$C_b= -(B_l)'$

GIVEN NETWORK:-



ALGORITHM:-

START

STEP1:

Form data matrices such that they represent line data of the given network
For the above network line data = [element no, from node, to node]

STEP2:

Form a zero matrix of order element by nodes

STEP3:

Use for loop and for each element choose the from node and to node in the created matrix and give the value as 1 and -1 respectively

STEP4:

Perform STEP3 for all elements to get the Bus incidence matrices

STEP5:

Remove the column corresponding to the reference node in the obtained matrix to get the reduced bus incidence matrices. Thus reduced incidence matrix A is formed.

STEP6

From the reduced incidence matrix select the rows which belong to tree elements and place them in matrix "A_b" and the rows which belong to link elements in matrix "A_l" .This way we can partition the A matrix into A_b and A_l

STEP7

Use the following relations to get the required matrices

$K_t = A_b^{-1}$ $\quad\quad$ $K = K_t'$

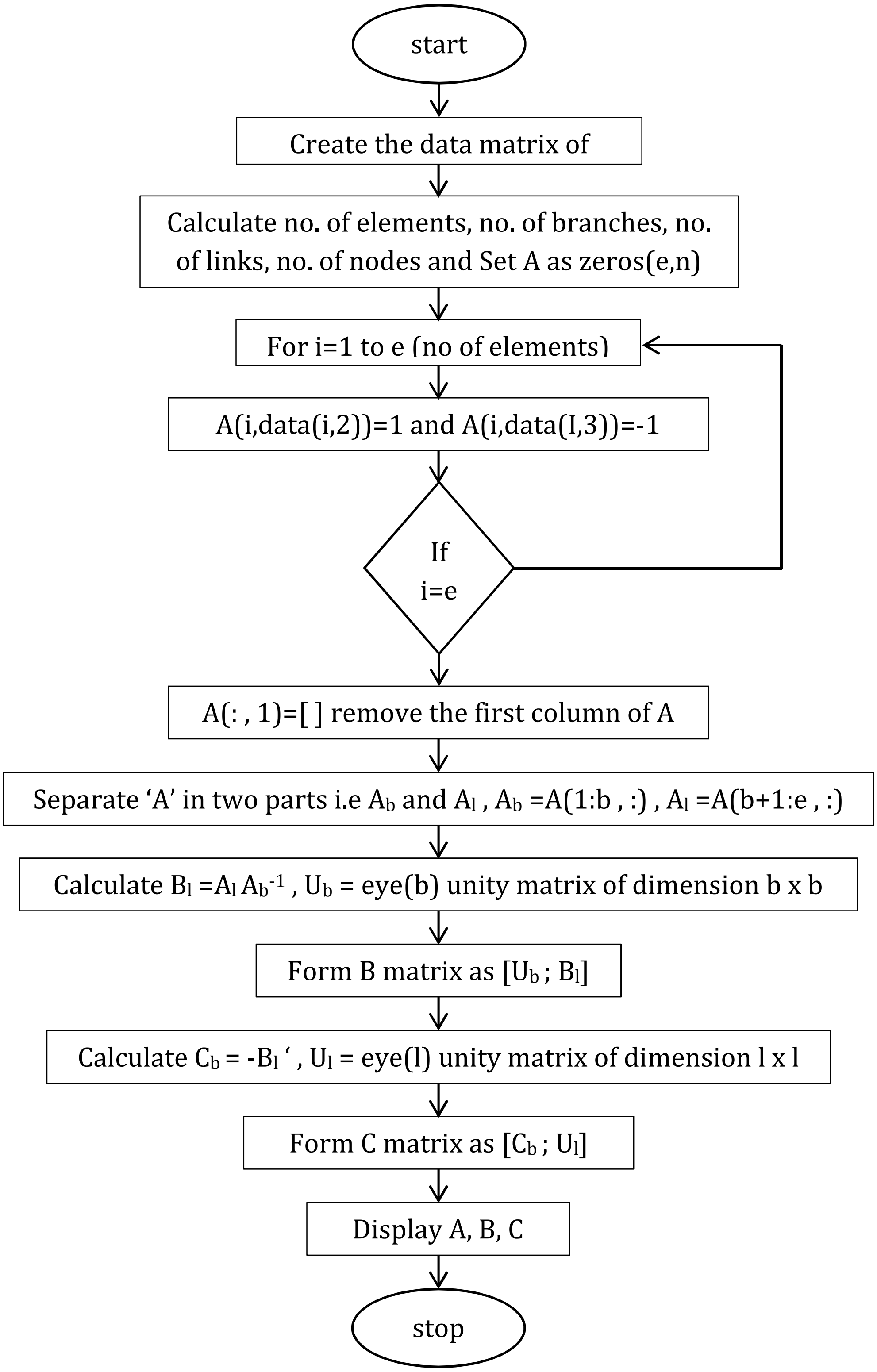
$B_l = A_l K_t$ $\quad\quad$ $C_b = -B_l'$

STEP8

Form augmented matrices of B and C

STOP

Flow chart:


start
Create the data matrix of
Calculate no. of elements, no. of branches, no. of links, no. of nodes and Set A as zeros(e,n)
For i=1 to e (no of elements)
A(i,data(i,2))=1 and A(i,data(I,3))=-1
If i=e
A(: , 1)=[ ] remove the first column of A
Separate 'A' in two parts i.e A_b and A_l , A_b =A(1:b , :) , A_l =A(b+1:e , :)
Calculate $B_l = A_l A_b^{-1}$, U_b = eye(b) unity matrix of dimension b x b
Form B matrix as [U_b ; B_l]
Calculate $C_b = -B_l$ ' , U_l = eye(l) unity matrix of dimension l x l
Form C matrix as [C_b ; U_l]
Display A, B, C
stop

PROGRAM:-

```
data=[1 1 2;2 1 3;3 1 4;4 5 1;5 2 3;6 4 3;7 5 4;8 2 3;9 3 4;10 4 5];
e=max(data(:,1));
n=max(max(data(:,2)),max(data(:,3)));
b=n-1;
l=e-b;
A=zeros(e,n);
for i=1:e
    A(i,data(i,2))=1;
    A(i,data(i,3))=-1;
end
A                %A is complete incidence matrix
A(:,1)=[]        % A is incidence matrix
Ab=A(1:b,:);
Al=A(b+1:e,:);
Bl=Al*Ab^(-1);
Ub=eye(b);
B=[Ub;Bl];
B                % B is basic cutset matrix
Cb=-(Bl)';
Ul=eye(l);
C=[Cb;Ul];
C                % C is basic loop matrix
```

RESULT:-

A =

-1	0	0	0
0	-1	0	0
0	0	-1	0
0	0	0	1
1	-1	0	0
0	-1	1	0
0	0	-1	1
1	-1	0	0
0	1	-1	0

B =

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
-1	1	0	0
0	1	-1	0
0	0	1	1
-1	1	0	0
0	-1	1	0
0	0	-1	-1

```
C =

     1     0     0     1     0     0
    -1    -1     0    -1     1     0
     0     1    -1     0    -1     1
     0     0    -1     0     0     1
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

PROGRAM – 3

Q) Write a general computer program to develop Y_{BUS}, Y_{BR} and Z_{LOOP} using singular transformation (no mutual coupling)

AIM:-

To determine Y Bus and Z Bus using singular transformation

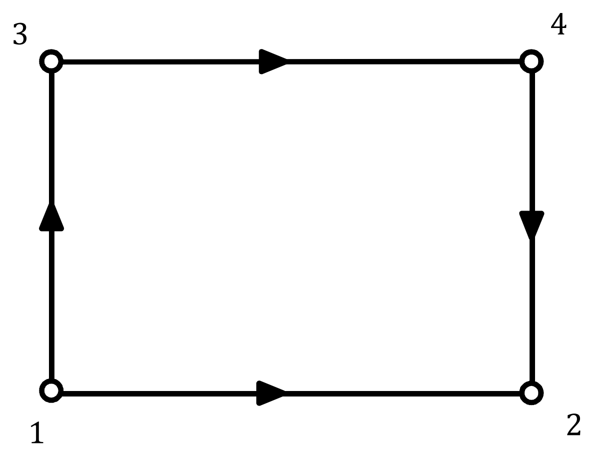
THEORY:-

By using primitive admittance matrices we can find Y bus matrices. The relation as follows

$$Y_{BUS} = A^t y A$$

Where "y" is primitive admittance matrix and "A" is reduced bus incidence matrix

GIVEN NETWORK:-



Element	From Bus	To Bus	Impedance	Tree
1	1	2	0.5	1
2	1	3	0.6	1
3	3	4	0.4	1
4	2	4	0.3	0

Mutual impedance data

Element 1	Element 2	Mutual impedance
2	1	0.1

ALGORITHM:-

STEP1

Create two matrices to represent line data and the mutual impedance data line data contains information about [element number, from node , to node, element impedance] whereas mutual data contains [element, mutual element, mutual impedance].

STEP2

Form the reduced bus incidence matrix A.

STEP3

Form the primitive impedance matrix 'z'. Where z is a matrix of order element by element in which diagonal elements are self-impedances and off diagonal elements are mutual impedances

STEP4

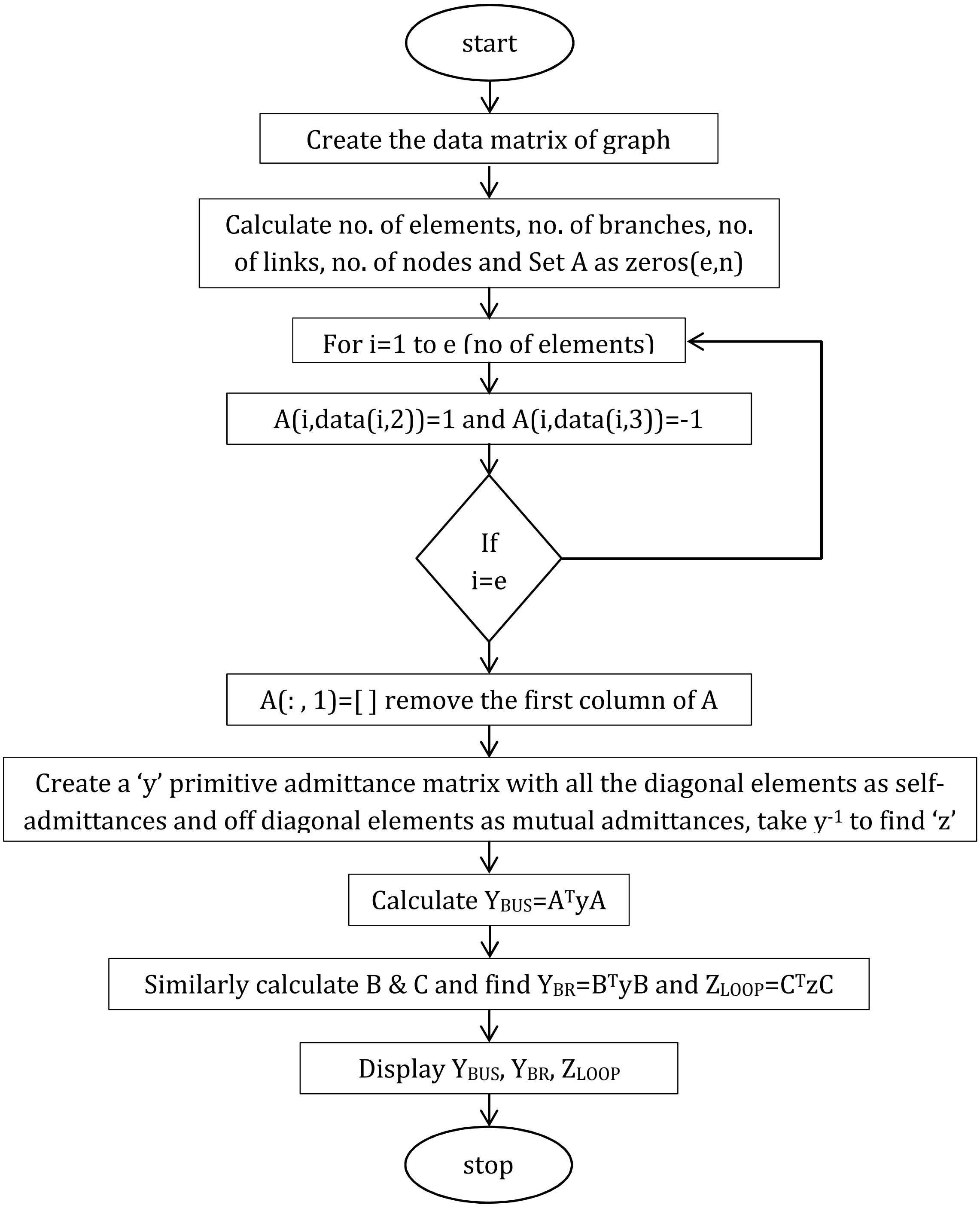
Find primitive admittance matrix 'y' using the relation $y = z^{-1}$

STEP5

Using the given relation $Y_{BUS} = A^t y A$ find Y_{BUS}

STOP

Flow chart:


start
Create the data matrix of graph
Calculate no. of elements, no. of branches, no. of links, no. of nodes and Set A as zeros(e,n)
For i=1 to e (no of elements)
A(i,data(i,2))=1 and A(i,data(i,3))=-1
If i=e
A(: , 1)=[ ] remove the first column of A
Create a 'y' primitive admittance matrix with all the diagonal elements as self-admittances and off diagonal elements as mutual admittances, take y^{-1} to find 'z'
Calculate $Y_{BUS}=A^TyA$
Similarly calculate B & C and find $Y_{BR}=B^TyB$ and $Z_{LOOP}=C^TzC$
Display Y_{BUS}, Y_{BR}, Z_{LOOP}
stop

PROGRAM:-

```
clc;
clear all;
data=[1 1 2 0.5;3 1 3 0.6;2 3 4 0.4;4 2 4 0.3];
mutual=[2 1 0.1];
m=length(mutual(:,1));        %no of mutual elements
z=diag(data(:,4));
for i=1:m
    z(mutual(i,1),mutual(i,2))=mutual(i,3);
    z(mutual(i,2),mutual(i,1))=mutual(i,3);
end
y=z^-1;
e=max(data(:,1));
n=max(max(data(:,2)),max(data(:,3)));
A=zeros(e,n);
for i=1:e
    A(i,data(i,2))=1;
    A(i,data(i,3))=-1;
end
A(:,1)=[]
Ybus=A'*y*A   %bus admittance matrix
Zbus=Ybus^-1  %bus impedance matrix
```

RESULT:-

```
A =


    -1      0      0

     0     -1      0

     0      1     -1

     1      0     -1




Ybus =


    5.4023   -0.3448   -3.3333

   -0.3448    4.2241   -2.5000

   -3.3333   -2.5000    5.8333
```

```
Zbus =

    0.4000    0.2250    0.3250
    0.2250    0.4437    0.3187
    0.3250    0.3187    0.4937
```

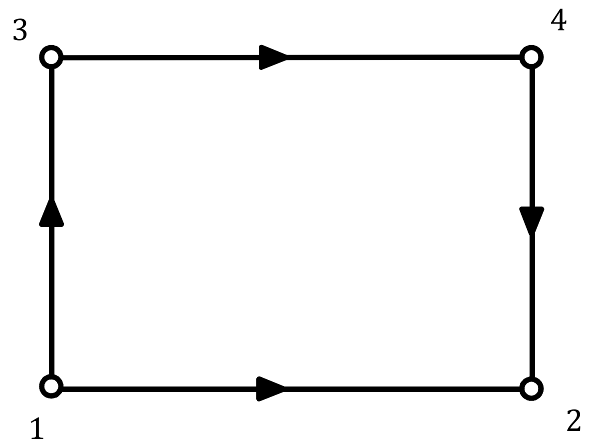
PROGRAM – 4

Q) Write a general computer program to develop Z_{BUS} by using Z_{BUS} building algorithm using addition of both link and branch

AIM:-

To find Z bus using Z bus building algorithm in MATLAB

GIVEN NETWORK:-



Element	From Bus	To Bus	Impedance	Tree
1	1	2	0.5	1
2	1	3	0.6	1
3	3	4	0.4	1
4	2	4	0.3	0

Mutual impedance data

Element 1	Element 2	Mutual impedance
2	1	0.1

ALGORITHM:-

Step1:

Create a data matrix with five columns, i.e., 'element no', 'from bus no' i.e. 'p' node, 'to bus no' i.e. 'q' node, 'self-admittance value', the fifth column consists of only 0 or 1. '1' if the element is a branch, '0' if the element is a link.

Step2:

The following conditions are to be applied while creating the data matrix

1. In the data matrix, the element numbering is given in a serial i.e. as the values of the element no. column proceeds numbers should go as 1, 2, 3 ….. etc.
2. We have to assume the procedure in the building algorithm as first adding the branches and then adding links. Therefore, in the fifth column we can find 1's first and then 0's.
3. The second column is always considered as 'p' node and while filling second and third column elements row-wise see that the second column element, is already present in the any of the 2^{nd} or 3^{rd} column elements in the above rows.

Step3:

Create a mutual matrix having three columns i.e. element 1, element 2, mutual admittance value.

Step4:

Calculate the primitive admittance matrix with all the diagonal elements as self-admittance values and off diagonal elements as the mutual admittance values (use the above data and mutual matrices). Find primitive impedance matrix by inversing the primitive admittance matrix

Step5:

Initialize 'Z' (bus impedance) as '0', qq (no of nodes added till now) as 1, ref (reference) as 1. Find no of elements 'e' and no of mutual elements 'm' using the data and mutual matrices.

Step6:

Start the 'for loop' with i=1 to e (no of elements). Take the corresponding second and third column elements as 'p' and 'q'. Initialize mutual1 (representing the existence of mutual element or not) with 0.

Step7:

Check if i=1, if yes then mutual1=0 because only one element has been added till now.

Step8:

Else initialize c=0 (count of the no of elements having mutual with the added element) after initializing check the pair of added element and the remaining each element added till now whether present in any of the rows in mutual matrix. If the pair is present in any row, then increase c by 1, r (rho) and s (sigma) as the corresponding 2^{nd} and 3^{rd} column values of that particular element (other than the added element in the pair found). Then make mutual1 as 1.

Step9:

Check If the added element is a branch or not. This can be checked by looking at the value of the fifth column of the data matrix. i.e. 1 for branch and 0 for link

Step10:

If it is a branch then make qq=qq+1 (increasing the count), increase the dimensions of the Z matrix by 1, i.e. one row and column has to be increased. Initialize the added elements with 0.

Step11:

Check if mutual1 is 1 or 0. If it is 1 then Check if p is reference or not. if yes then apply the corresponding calculation of finding the Z_{qi} and Z_{qq}. Similarly if p is not a reference then find corresponding Z_{qi} and Z_{qq} values. Do the same (i.e corresponding formulas for finding Z_{qi} and Z_{qq}) if mutual1 is 0.

Step12:

If the added element is a link, then ll=qq+1, increase the dimension of the Z matrix by 1x1 temporarily, i.e. one row and one column has to be increased. Initialize the added elements with 0.

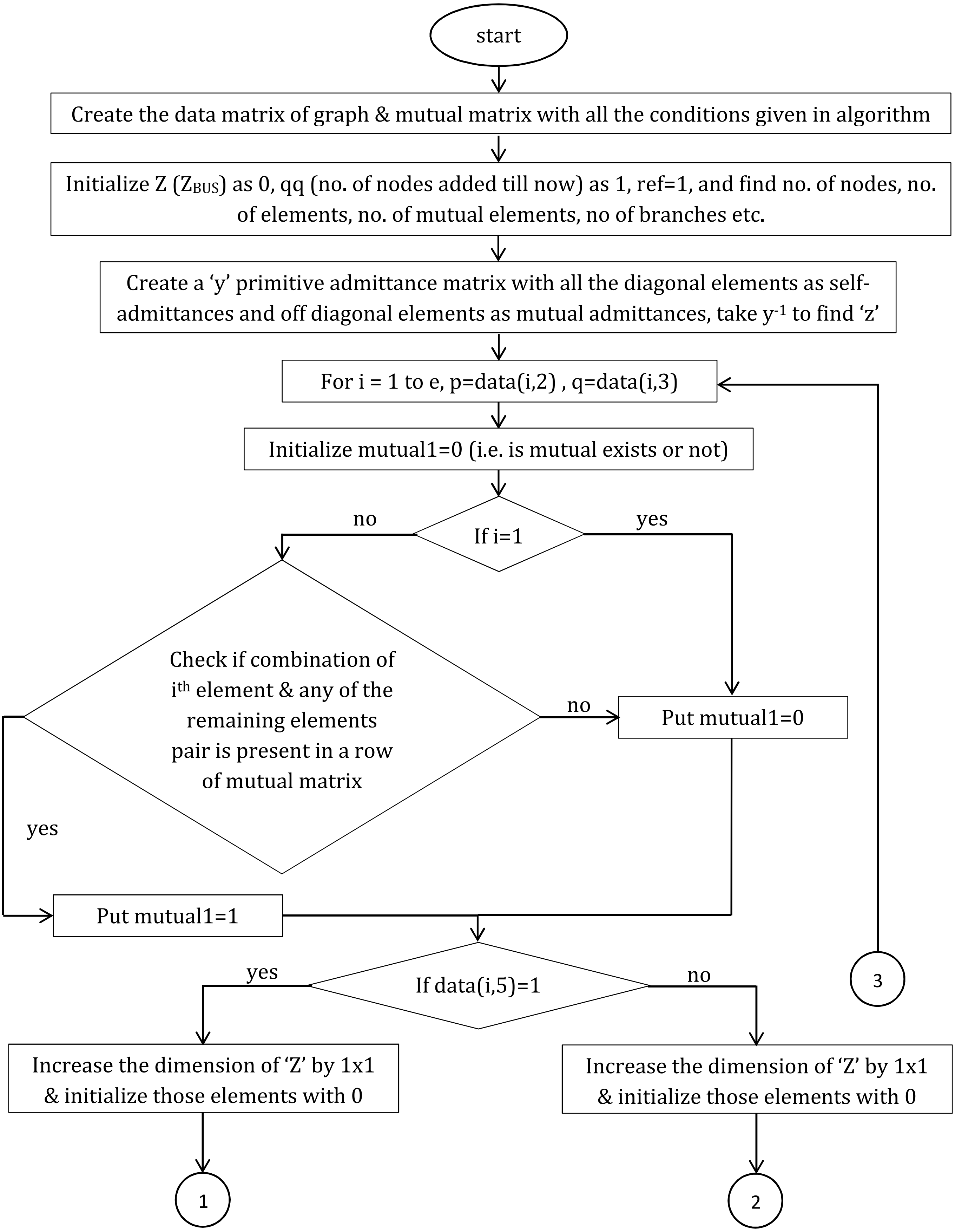
Step13:

Check if mutual1 is 1 or 0. If it is 1 then Check if p is reference or not. if yes then apply the corresponding calculation of finding the Z_{li} and Z_{ll}. Similarly if p is not a reference then find corresponding Z_{li} and Z_{ll} values. Do the same (i.e corresponding formulas for finding Z_{li} and Z_{ll}) if mutual1 is 0. Immediately modify the Z matrix values except the link related values and remove the link related row and column, reducing the dimension of Z.

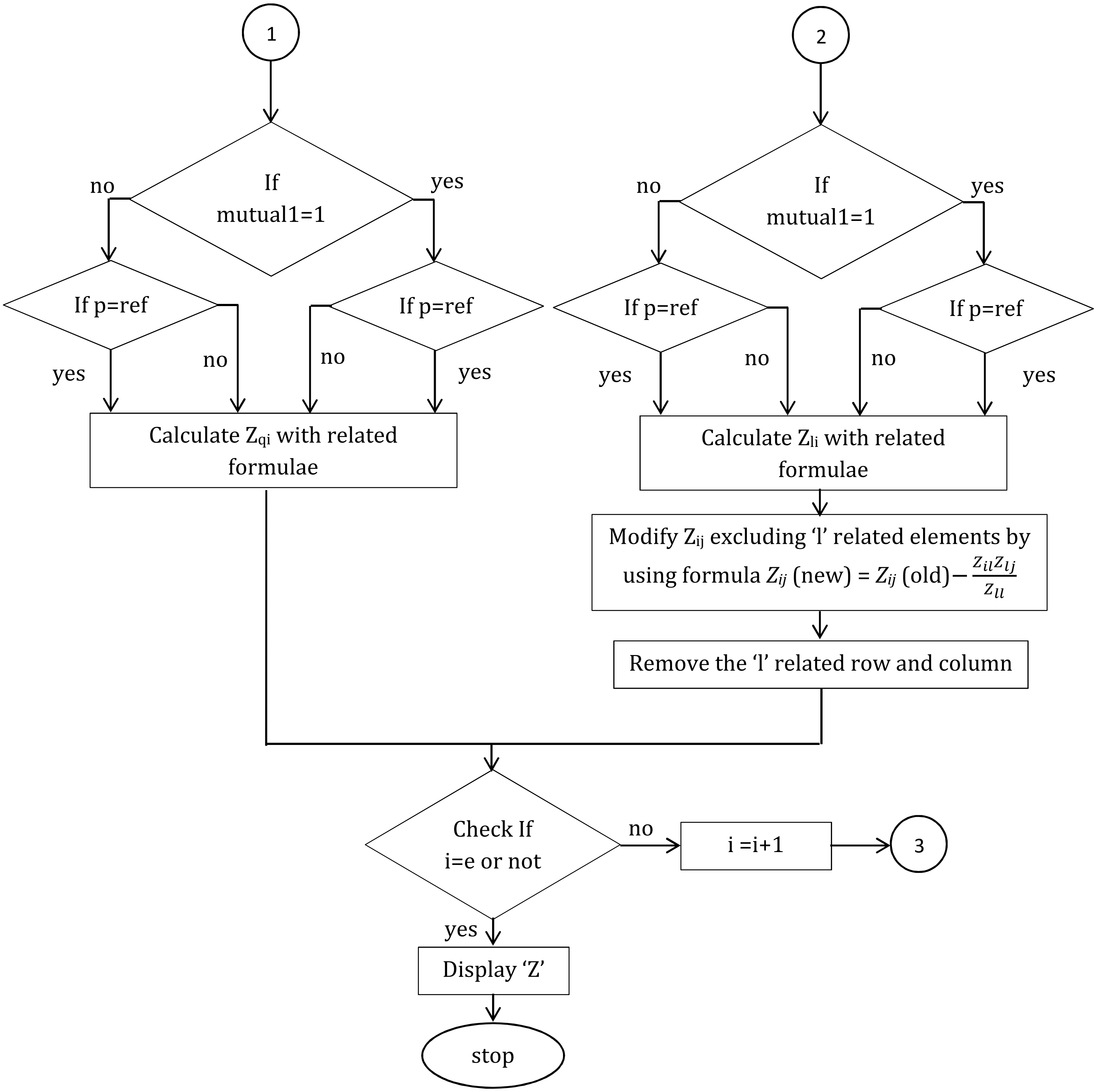
Step14:

Finally the i value comes to e and the loop ends. To display the final Z matrix simply type Z without any semicolon.

Flow chart:



start
Create the data matrix of graph & mutual matrix with all the conditions given in algorithm
Initialize Z (ZBUS) as 0, qq (no. of nodes added till now) as 1, ref=1, and find no. of nodes, no. of elements, no. of mutual elements, no of branches etc.
Create a 'y' primitive admittance matrix with all the diagonal elements as self-admittances and off diagonal elements as mutual admittances, take y-1 to find 'z'
For i = 1 to e, p=data(i,2) , q=data(i,3)
Initialize mutual1=0 (i.e. is mutual exists or not)
If i=1
no
yes
Check if combination of ith element & any of the remaining elements pair is present in a row of mutual matrix
no
Put mutual1=0
yes
Put mutual1=1
If data(i,5)=1
yes
no
Increase the dimension of 'Z' by 1x1 & initialize those elements with 0
Increase the dimension of 'Z' by 1x1 & initialize those elements with 0
1
2
3


1
If
mutual1=1
no
yes
If p=ref
If p=ref
yes
no
no
yes
Calculate Z_{qi} with related formulae
2
If
mutual1=1
no
yes
If p=ref
If p=ref
yes
no
no
yes
Calculate Z_{li} with related formulae
Modify Z_{ij} excluding 'l' related elements by using formula Z_{ij} (new) = Z_{ij} (old)$-\frac{Z_{il}Z_{lj}}{Z_{ll}}$
Remove the 'l' related row and column
Check If
i=e or not
no
i =i+1
3
yes
Display 'Z'
stop

PROGRAM:-

```
clc;
clear all;
data=[1 1 2 0.5 1;3 1 3 0.6 1;2 3 4 0.4 1;4 2 4 0.3 0];
mutual=[2 1 0.1]; %mutual impedance data (impedance zm between
e1 and e2)
e=length(data(:,1));           %no of elements
Z=0;
qq=1;
m=length(mutual(:,1));
ref=data(1,1);                           %reference node
z=diag(data(:,4));
for i=1:m
    z(mutual(i,1),mutual(i,2))=mutual(i,3);
    z(mutual(i,2),mutual(i,1))=mutual(i,3);
end
y=z^-1;
for i=1:e
    p=data(i,2);
    q=data(i,3);
    mutual1=0;
    if i==1
        mutual1=0;
    else
        c=0;
        for i1=1:i-1
            for ii=1:m
    if ismember(i,mutual(ii,1:2))==1&&ismember(i1,mutual(ii,1:2))==1
                c=c+1;
                v(c)=i1;
                r(c)=data(i1,2);
                s(c)=data(i1,3);
                mutual1=1;
            end
            end
        end
    end
    if data(i,5)==1
        qq=qq+1;
        zz=zeros(1,qq-1);
        Z=[Z zz';zz zeros(1)];
        if mutual1==1
            if p==ref
                for ii=1:qq
                    if ii~=q&&ii~=ref
                      for k=1:c
                      g(c)=y(i,v(c))*(Z(r(c),ii)-Z(s(c),ii));
                      end
                        Z(q,ii)=sum(g)/y(i,i);
                        Z(ii,q)=Z(q,ii);
```

```
                end
                for k=1:c
                    g1(c)=y(i,v(c))*(Z(r(c),q)-Z(s(c),q));
                end
            end
            Z(q,q)=(1+sum(g1))/y(i,i);
        else
            for ii=1:qq
                if ii~=q&&ii~=ref
                for k=1:c
                g(c)=y(i,v(c))*(Z(r(c),ii)-Z(s(c),ii));
                end
                    Z(q,ii)=Z(p,ii)+sum(g)/y(i,i);
                    Z(ii,q)=Z(q,ii);
                end
                for k=1:c
                    g1(c)=y(i,v(c))*(Z(r(c),q)-Z(s(c),q));
                end
            end
            Z(q,q)=Z(p,q)+(1+sum(g1))/y(i,i);
        end
    else
        if p==ref
            Z(q,q)=z(i,i);
        else
            for ii=1:qq
                if ii~=q&&ii~=ref
                    Z(q,ii)=Z(p,ii);
                    Z(ii,q)=Z(q,ii);
                end
            end
            Z(q,q)=Z(p,q)+z(i,i);
        end
    end
else
    ll=qq+1;
    zz1=zeros(1,qq);
    Z=[Z zz1';zz1 zeros(1)];
    if mutual1==1
        if p==ref
            for ii=1:qq
                if ii~=ll&&ii~=ref
                   for k=1:c
                   g(c)=y(i,v(c))*(Z(r(c),ii)-Z(s(c),ii));
                   end
                    Z(ll,ii)=-Z(q,ii)+sum(g)/y(i,i);
                    Z(ii,ll)=Z(ll,ii);
                end
                 for k=1:c
                  g1(c)=y(i,v(c))*(Z(r(c),ll)-Z(s(c),ll));
                 end
```

```
                end
                Z(ll,ll)=-Z(q,ll)+(1+sum(g1))/y(i,i);
            else
                for ii=1:qq
                    if ii~=ll&&ii~=ref
                      for k=1:c
                       g(c)=y(i,v(c))*(Z(r(c),ii)-Z(s(c),ii));
                      end
                       Z(ll,ii)=Z(p,ii)-Z(q,ii)+sum(g)/y(i,i);
                       Z(ii,ll)=Z(ll,ii);
                    end
                    for k=1:c
                      g1(c)=y(i,v(c))*(Z(r(c),ll)-Z(s(c),ll));
                    end
                end
                Z(ll,ll)=Z(p,ll)-Z(q,ll)+(1+sum(g1))/y(i,i);
            end
        else
            if p==ref
                for ii=1:qq
                    if ii~=ll&&ii~=ref
                        Z(ll,ii)=-Z(q,ii);
                        Z(ii,ll)=Z(ll,ii);
                    end
                end
                Z(ll,ll)=-Z(q,ll)+z(i,i);
            else
                for ii=1:qq
                    if ii~=ll&&ii~=ref
                        Z(ll,ii)=Z(p,ii)-Z(q,ii);
                        Z(ii,ll)=Z(ll,ii);
                    end
                end
                Z(ll,ll)=Z(p,ll)-Z(q,ll)+z(i,i);
            end
        end
        for ii=1:qq
            for jj=1:qq
                Z(ii,jj)=Z(ii,jj)-Z(ii,ll)*Z(ll,jj)/Z(ll,ll);
            end
        end
        Z=Z(1:qq,1:qq);
    end
end
Z
```

Result:-

```
Z =

     0         0         0         0
     0    0.4000    0.2250    0.3250
     0    0.2250    0.4437    0.3187
     0    0.3250    0.3187    0.4938
```

PROGRAM:-5

Q) Write a generalized program to find the fault current and voltage profile of 3 phase bus system when LLLG fault occurs at any bus.

AIM:-

To find voltage and current after the occurrence of LLLG fault at buses

ALGORITHM:-

START

STEP1:

Enter the enter the no. of 3-phase buses in the power system, Z_f, Z_g, fault occurred bus values

STEP2:

Enter the positive sequence impedance value of Z_{pi}

STEP3:

Take a=-0.5+$\frac{\sqrt{3}}{2}i$ for further calculations

STEP4:

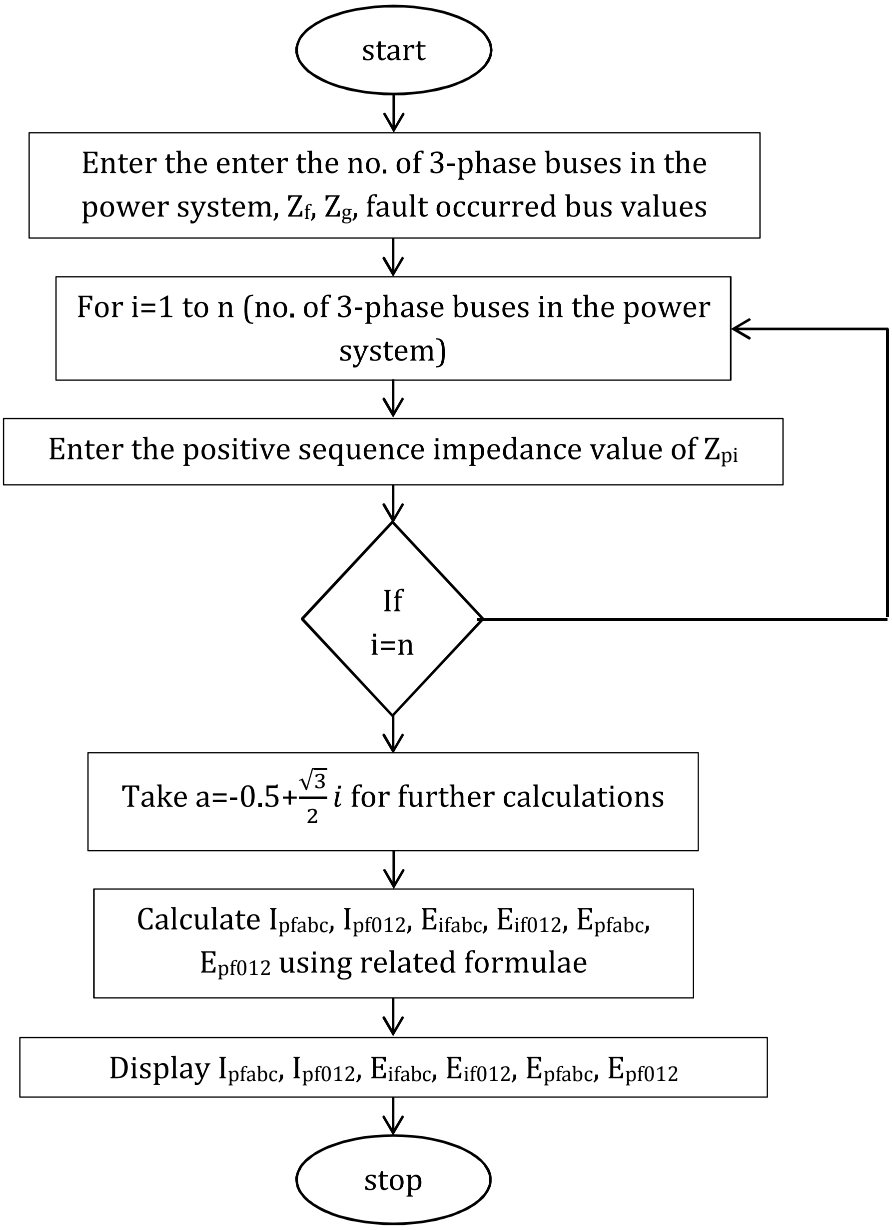
Calculate I_{pfabc}, I_{pf012}, E_{ifabc}, E_{if012}, E_{pfabc}, E_{pf012} using related formulae

STEP5:

If all the bus voltage and bus currents calculations are completed then display the results

STOP

Flow chart:



Program:-

```
clc;
clear all;
disp('LLLG fault current and voltage calculations = ');
n=input('enter the no. of 3-phase buses in the power system =
');
p=input('the LLLG fault has occured at which bus = ');
Zf=input('enter the Zf value = ');
Zg=input('enter the Zg value = ');
```

```
for i=1:n
    Zp(i)=input(sprintf('enter the positive sequence impedence value of Z%d%d = ',p,i));
end
a=(-0.5)+(sqrt(3)/2)*i;
Ipfabc=(1/(Zf+Zp(p)))*[1;a^2;a];
Ipf012=(1/(Zf+Zp(p)))*[0;sqrt(3);0];
Eifabc(:,p)=(Zf/(Zf+Zp(p)))*[1;a^2;a];
Eif012(:,p)=(Zf/(Zf+Zp(p)))*[0;sqrt(3);0];
for i=1:n
    if i~=p
        Eifabc(:,i)=(1-Zp(i)/(Zf+Zp(p)))*[1;a^2;a];
        Eif012(:,i)=(1-Zp(i)/(Zf+Zp(p)))*[0;sqrt(3);0];
    end
end
Epfabc=Eifabc(:,p);
Epf012=Eif012(:,p);
Eifabc
Eif012
Epfabc
Epf012
Ipfabc
Ipf012
```

RESULT:-

```
LLLG fault current and voltage calculations =
enter the no. of 3-phase buses in the power system = 2
the LLLG fault has occured at which bus = 2
enter the Zf value = .3
enter the Zg value = .1
enter the positive sequence impedence value of Z21 = .4
enter the positive sequence impedence value of Z22 = .4

Eifabc =
   0.4286    0.4286
    0.6505    0.6505
    0.5280    0.5280

Eif012 =
         0         0
    0.7423    0.7423
         0         0

Epfabc =
    0.4286
    0.6505
```

```
    0.5280

Epf012 =
         0
    0.7423
         0

Ipfabc =
    1.4286
    2.1685
    1.7601

Ipf012 =
         0
    2.4744
         0
```