BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Project report

Course No : EEE 212 **Group No :** 06 (ID:1906011 and 1906012)

Course Title : Numerical Technique Laboratory

Section : A1

Level : 2 , **Term :** 1

Submitted To:

Dr. Md. Shah Alam	**Ishtiak Mahbub**
Professor, Department of EEE	**Lecturer (PT), Department of EEE,**
BUET	**BUET**

Submitted by:

Md.Sad Abdullah Sami

ID : 1906011

Shahan Bin Sharif Hirock

ID : 1906012

Project Name:

THREE-PHASE CIRCUIT SOLVER

Introduction:

Our country used to face frequent power shortages due to lack of power supply.Even in near past early 2000s poweroutage was horrible.In the early stage of previous decade we gradually improved our situation.Three-phase system played a big part in it.

Three-phase circuits were introduced instead of single-phase circuits for better power output.Three-phase circuit is a combination of three sources with 120 degree phase difference with each other.This sources are connected to three load impedences.The loss in the transmition line is normally considered as line impedence.These three-phase circuits have four combinations.They are:Y-delta,delta-Y,Y-Y,delta-delta. In this project we made a three-phase circuit solver.This circuit solver takes input about source,load and line impedence and gives us line,load currents,line,load voltages,real,complex,reactive power,power factor,phasor diagram etc. meaning all the data of interest in a three phase circuit.

Objectives and features:

The goal of this project is to provide a fully working three-phase circuit solver.For this some info is taken from user and then processed and some outputs are given.

Info taken from user:

1.What type of combination

2.Source voltages

3.Line impedences

4.Load impedences

Output given:

1.Line currents

2.Load currents

3.Line voltages

4.Load voltages

5.Total power

6.Reactive power

7.Real power

8.Power factor

9.Phase diagram of load and line voltages,load and line currents.

<u>Rough flowchart of the programme:</u>

Taking input from user --→ What combination ---→ Source voltages,Line

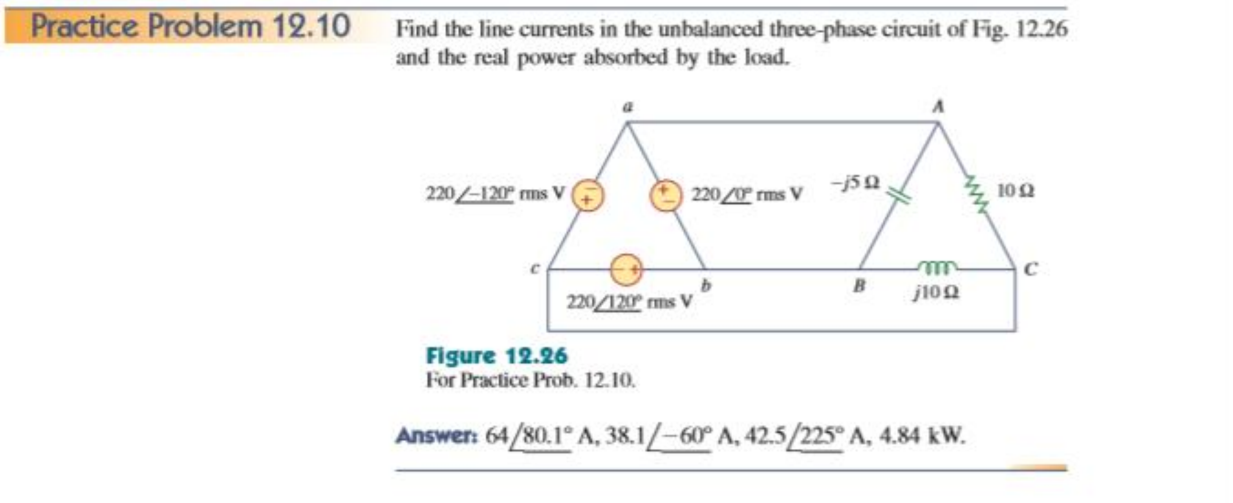
impedences,Load impedences --→ Using Mesh analysis -- -→

Finding Line currents,Load currents,Line voltages,Load voltages using the results from mesh and according to combination --→

Finding Total power,Reactive power,Real power,Power factor,Phase diagram of load and line voltages,load and line currents ---→

Give all output and figure

<u>Input from user:</u>



Practice Problem 12.10

Find the line currents in the unbalanced three-phase circuit of Fig. 12.26 and the real power absorbed by the load.

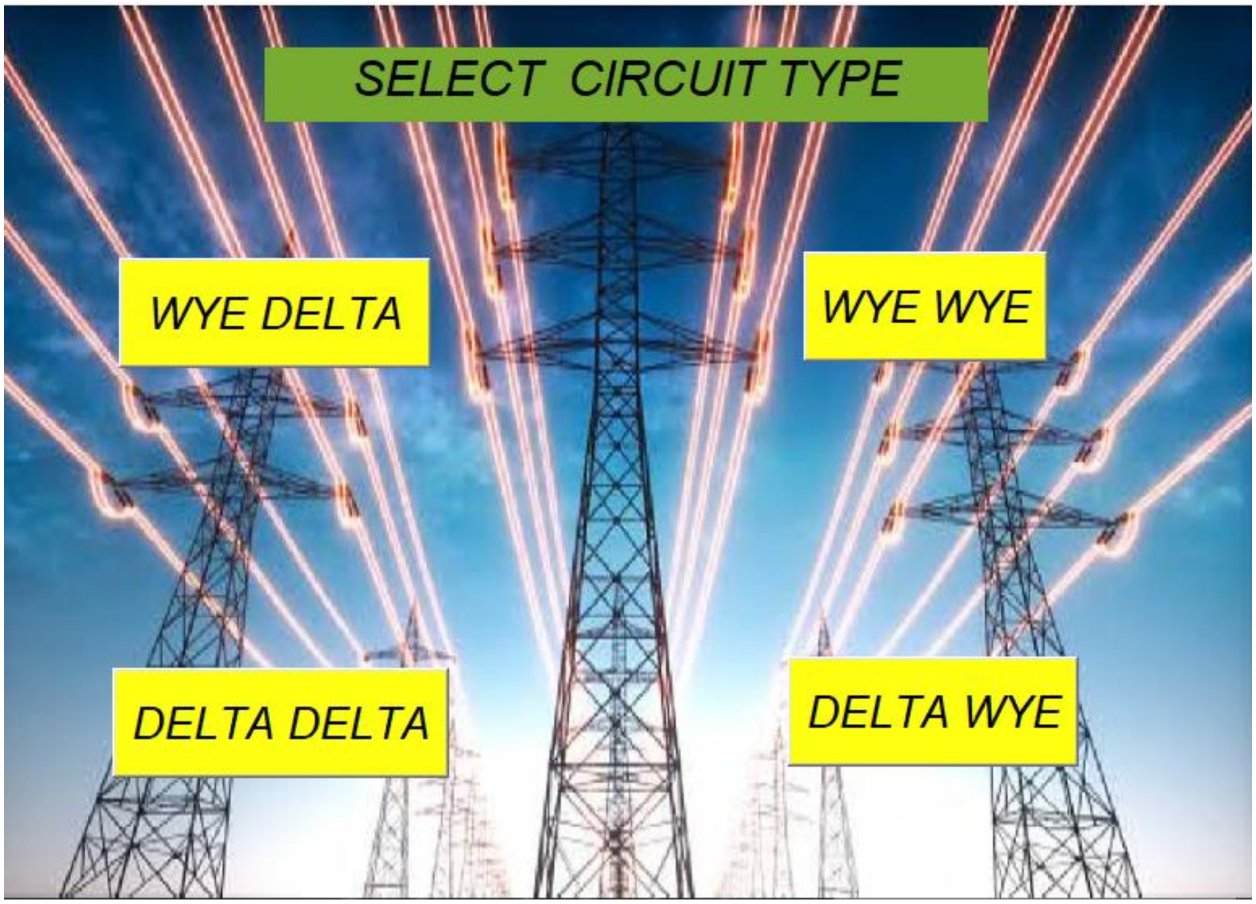
a
A
220/−120° rms V
220/0° rms V
−j5 Ω
10 Ω
c
b
B
j10 Ω
C
220/120° rms V

Figure 12.26
For Practice Prob. 12.10.

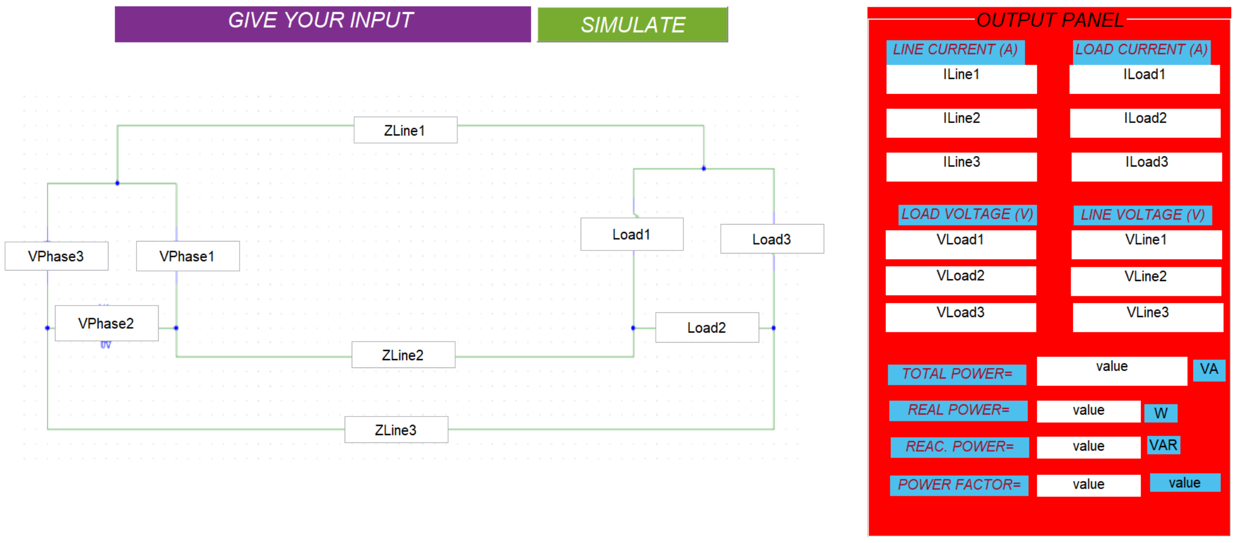
Answer: $64\angle 80.1^\circ$ A, $38.1\angle -60^\circ$ A, $42.5\angle 225^\circ$ A, 4.84 kW.

For taking input lets consider this example from sadiku 5th edition.Its a delta-delta unbalanced connection.

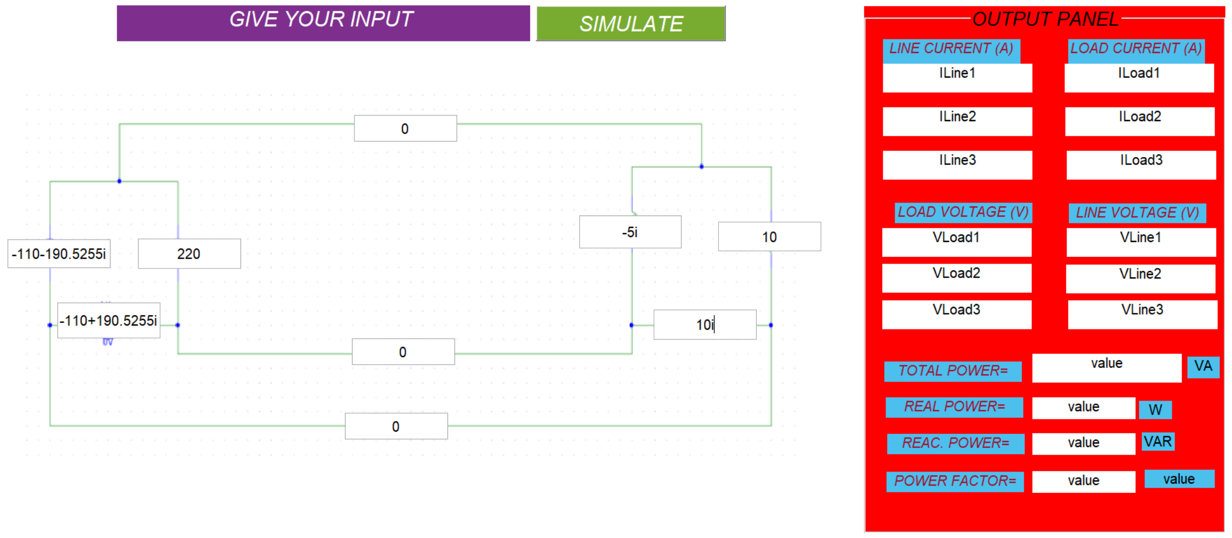
First user will have to choose delta-delta from menu gui.



After choosing delta delta a new GUI will appear.



In this gui the user will give the input values on the screen.



0

-110-190.5255i

220

-110+190.5255i

-5i

10

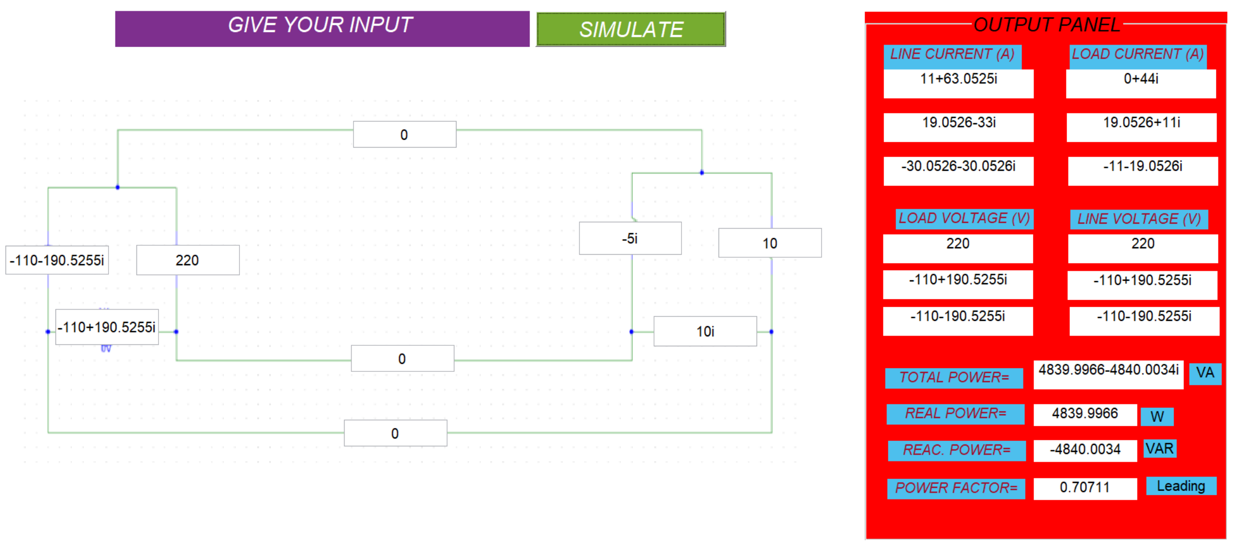
10i

0

0

After putting input the user will click “SIMULATE” and his job is done!

Output :



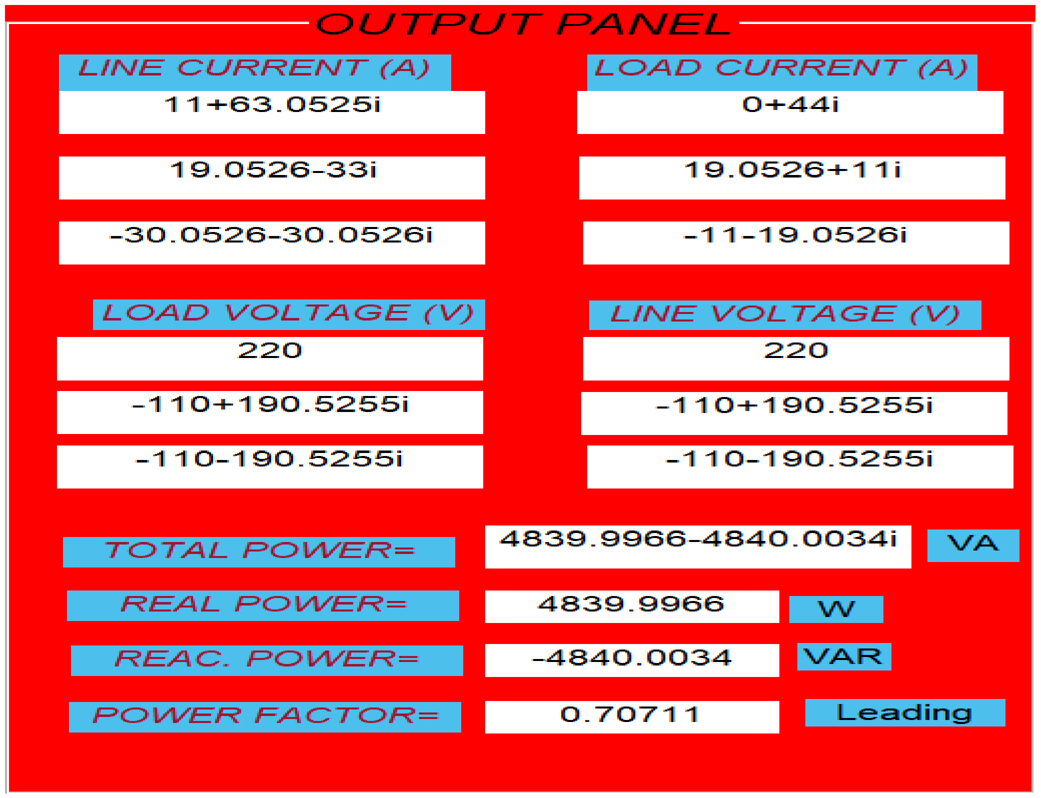
OUTPUT PANEL

LINE CURRENT (A)	LOAD CURRENT (A)
11+63.0525i	0+44i
19.0526-33i	19.0526+11i
-30.0526-30.0526i	-11-19.0526i

LOAD VOLTAGE (V)	LINE VOLTAGE (V)
220	220
-110+190.5255i	-110+190.5255i
-110-190.5255i	-110-190.5255i

TOTAL POWER=	4839.9966-4840.0034i	VA
REAL POWER=	4839.9966	W
REAC. POWER=	-4840.0034	VAR
POWER FACTOR=	0.70711	Leading

Output will be visible on “Output Panel”.Let’s zoom in the output and check whether or not it matches textbook.

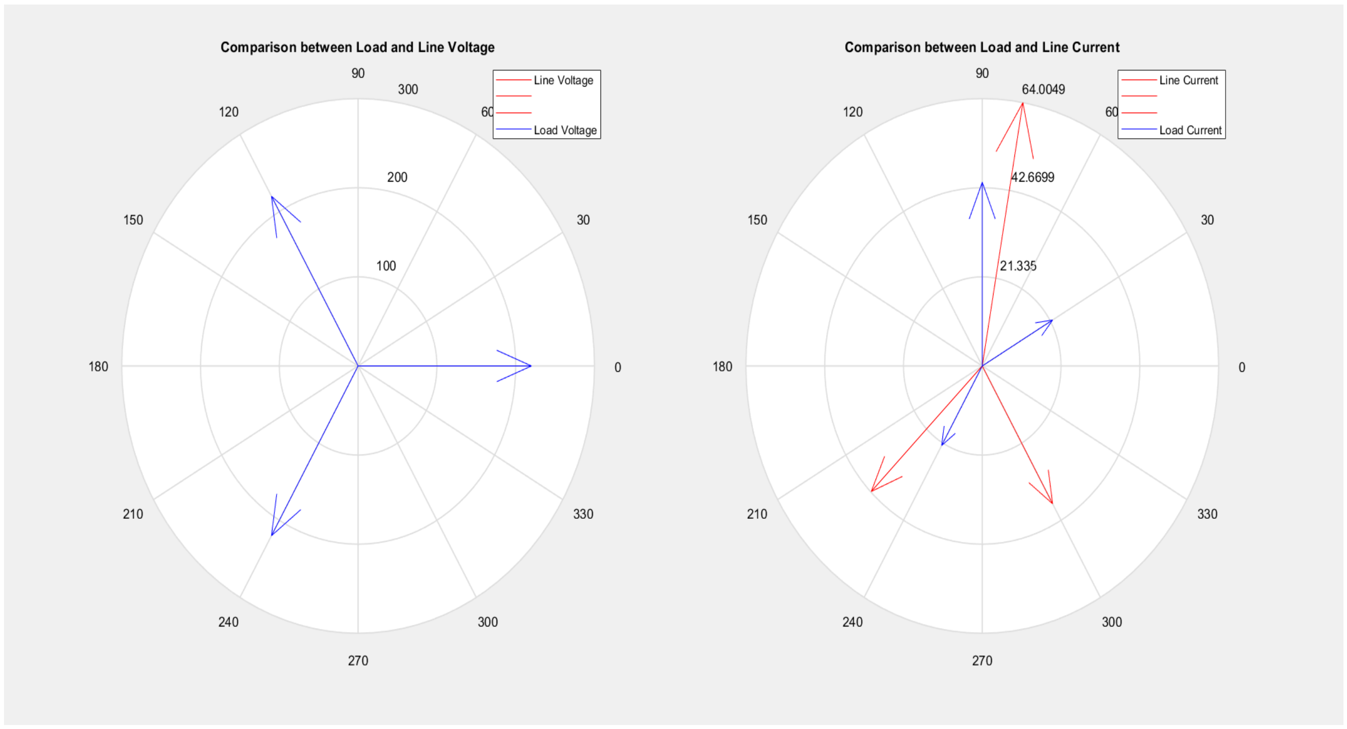


OUTPUT PANEL

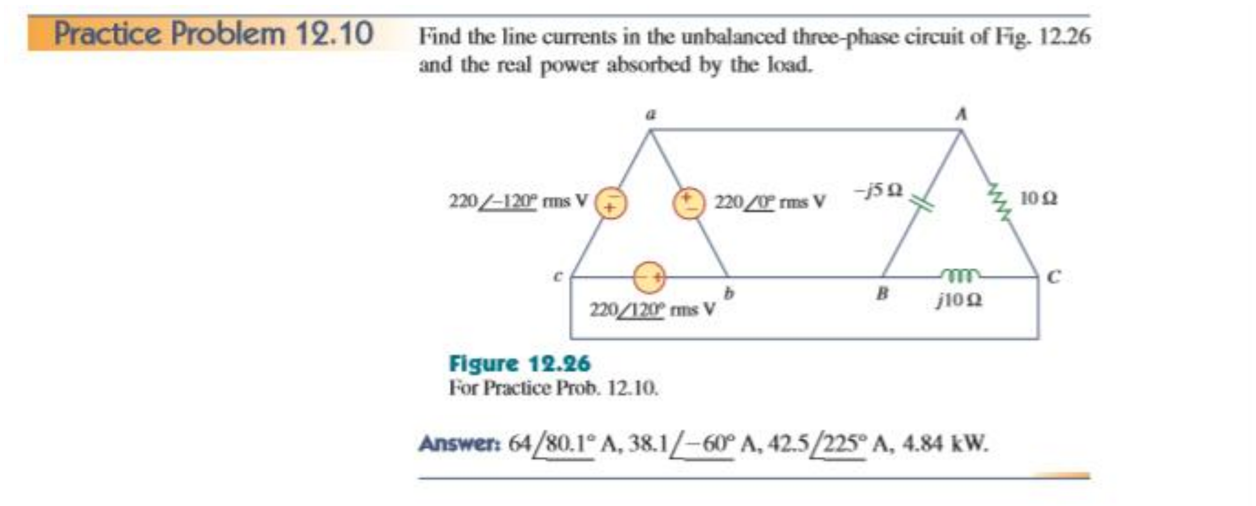
LINE CURRENT (A)	LOAD CURRENT (A)
11+63.0525i	0+44i
19.0526-33i	19.0526+11i
-30.0526-30.0526i	-11-19.0526i

LOAD VOLTAGE (V)	LINE VOLTAGE (V)
220	220
-110+190.5255i	-110+190.5255i
-110-190.5255i	-110-190.5255i

TOTAL POWER=	4839.9966-4840.0034i	VA
REAL POWER=	4839.9966	W
REAC. POWER=	-4840.0034	VAR
POWER FACTOR=	0.70711	Leading



This are phase diagrams of load and line voltages,load and line currents.



Practice Problem 12.10 Find the line currents in the unbalanced three-phase circuit of Fig. 12.26 and the real power absorbed by the load.

a
A
b
B
c
C
220/−120° rms V
220/0° rms V
220/120° rms V
−j5 Ω
10 Ω
j10 Ω

Figure 12.26
For Practice Prob. 12.10.

Answer: $64\angle 80.1^\circ$ A, $38.1\angle -60^\circ$ A, $42.5\angle 225^\circ$ A, 4.84 kW.

Now the real power is 4.84 kW in sadiku and in the output panel it is 4839.996

Line current is 64<80.1 = 64cos(80.1)+i64sin(80.1)

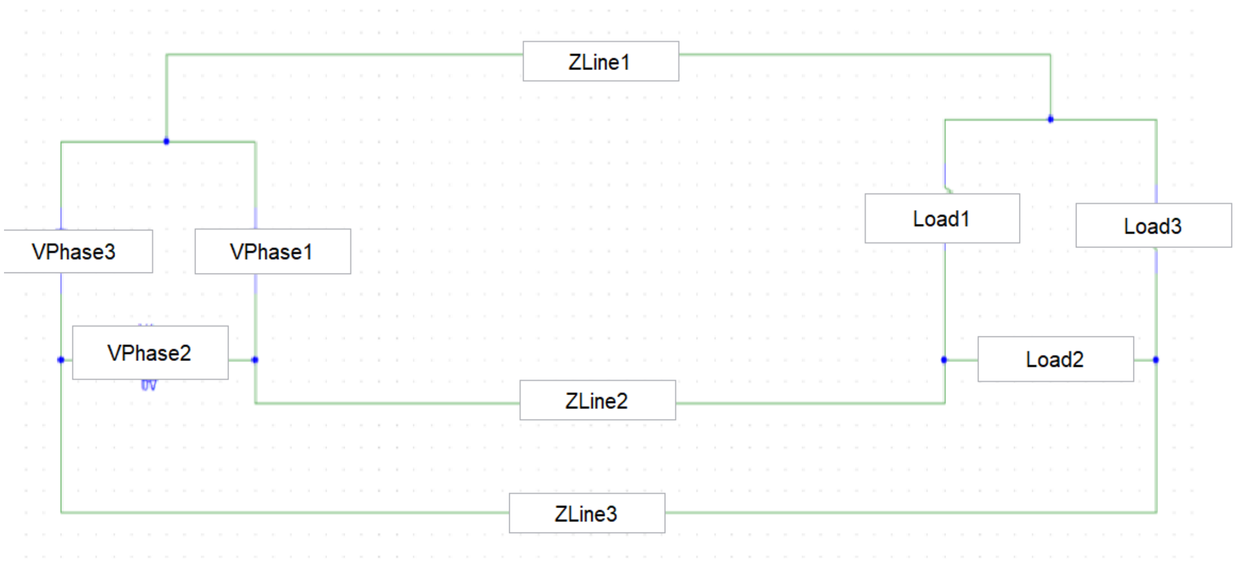
=11.0034+63.04i

Other line currents also match the answer of sadiku.

So,its safe to say the code gives output accurately.

Algorithm used for solving this problem:

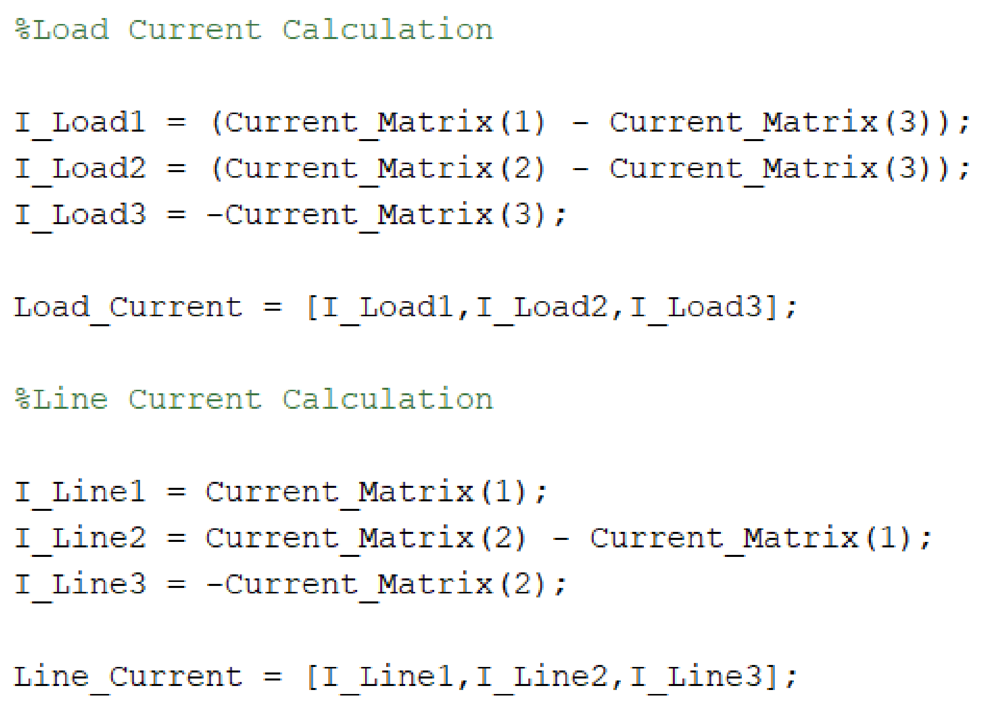
At first inputs were taken.



Using those inputs Mesh analysis was used.The reason for using mesh analysis is very simple.It can solve both balanced and unbalanced circuits.Two matrices were created.

```
Matrix1 = [(Z_Line1 + Load1 + Z_Line2) -(Z_Line2) -Load1;...
    -(Z_Line2) (Z_Line2 + Load2 + Z_Line3) -Load2;...
    -Load1 -Load2 (Load1 + Load2 + Load3)];

Matrix2 = [V_Phase1; V_Phase2; 0];

Current_Matrix = inv(Matrix1)*Matrix2;
```

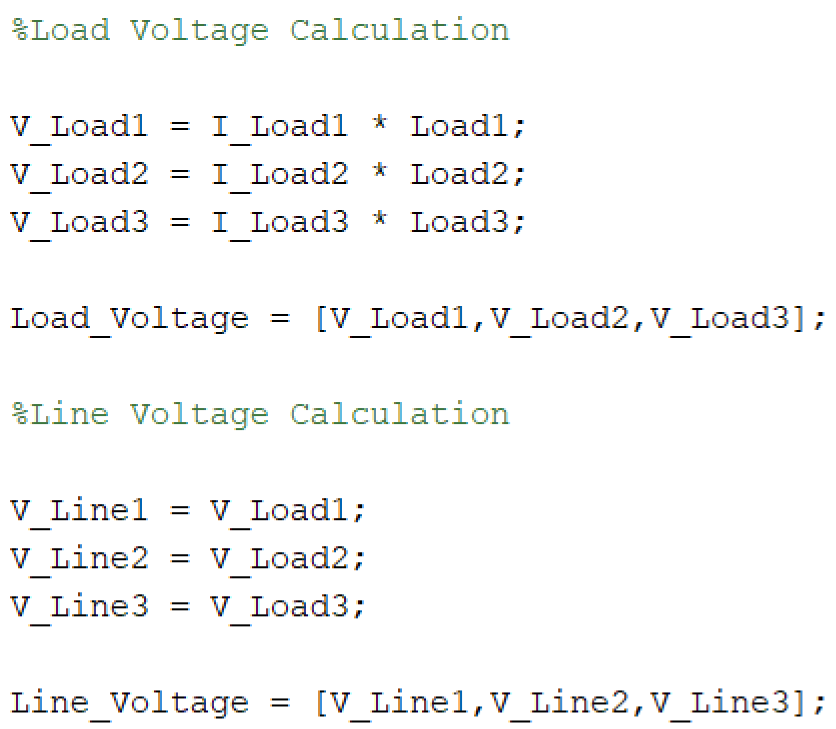
Matrix1 contains the co-effecients of currents in the mesh and matrix2 contains the constant parts.We inversed the first matrix and multiplied it with the second matrix to get the current matrix.



```
%Load Current Calculation

I_Load1 = (Current_Matrix(1) - Current_Matrix(3));
I_Load2 = (Current_Matrix(2) - Current_Matrix(3));
I_Load3 = -Current_Matrix(3);

Load_Current = [I_Load1,I_Load2,I_Load3];

%Line Current Calculation

I_Line1 = Current_Matrix(1);
I_Line2 = Current_Matrix(2) - Current_Matrix(1);
I_Line3 = -Current_Matrix(2);

Line_Current = [I_Line1,I_Line2,I_Line3];
```

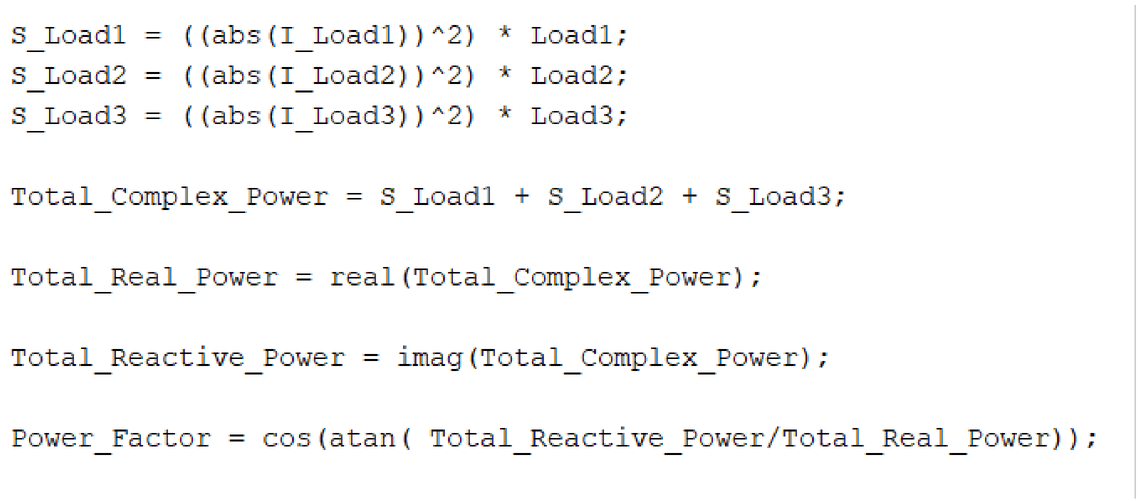
Using the current matrix that contains the mesh currents we got the load currents and line currents.

With the help of this currents and the line and load impedences provided by the user we got the line and load voltages.



```
%Load Voltage Calculation

V_Load1 = I_Load1 * Load1;
V_Load2 = I_Load2 * Load2;
V_Load3 = I_Load3 * Load3;

Load_Voltage = [V_Load1,V_Load2,V_Load3];

%Line Voltage Calculation

V_Line1 = V_Load1;
V_Line2 = V_Load2;
V_Line3 = V_Load3;

Line_Voltage = [V_Line1,V_Line2,V_Line3];
```

Now,for the power calculation part.



```
S_Load1 = ((abs(I_Load1))^2) * Load1;
S_Load2 = ((abs(I_Load2))^2) * Load2;
S_Load3 = ((abs(I_Load3))^2) * Load3;

Total_Complex_Power = S_Load1 + S_Load2 + S_Load3;

Total_Real_Power = real(Total_Complex_Power);

Total_Reactive_Power = imag(Total_Complex_Power);

Power_Factor = cos(atan( Total_Reactive_Power/Total_Real_Power));
```

First we calcultated the complex power for each load.We did that using,

S = |I|^2 * Zload ;   formula

After calculating for each load we simply added them to get the total complex power.

From the complex power we retrieved the real power using real(Total_complex_power) which gives the real value of complex power and the reactive power using imag(Total_complex_power) which gives the imaginary part of complex power.

For power factor calculation we used,

tan(theda)=Qtotal/Ptotal;

So theda = tan_inverse(Qtotal/Ptotal)

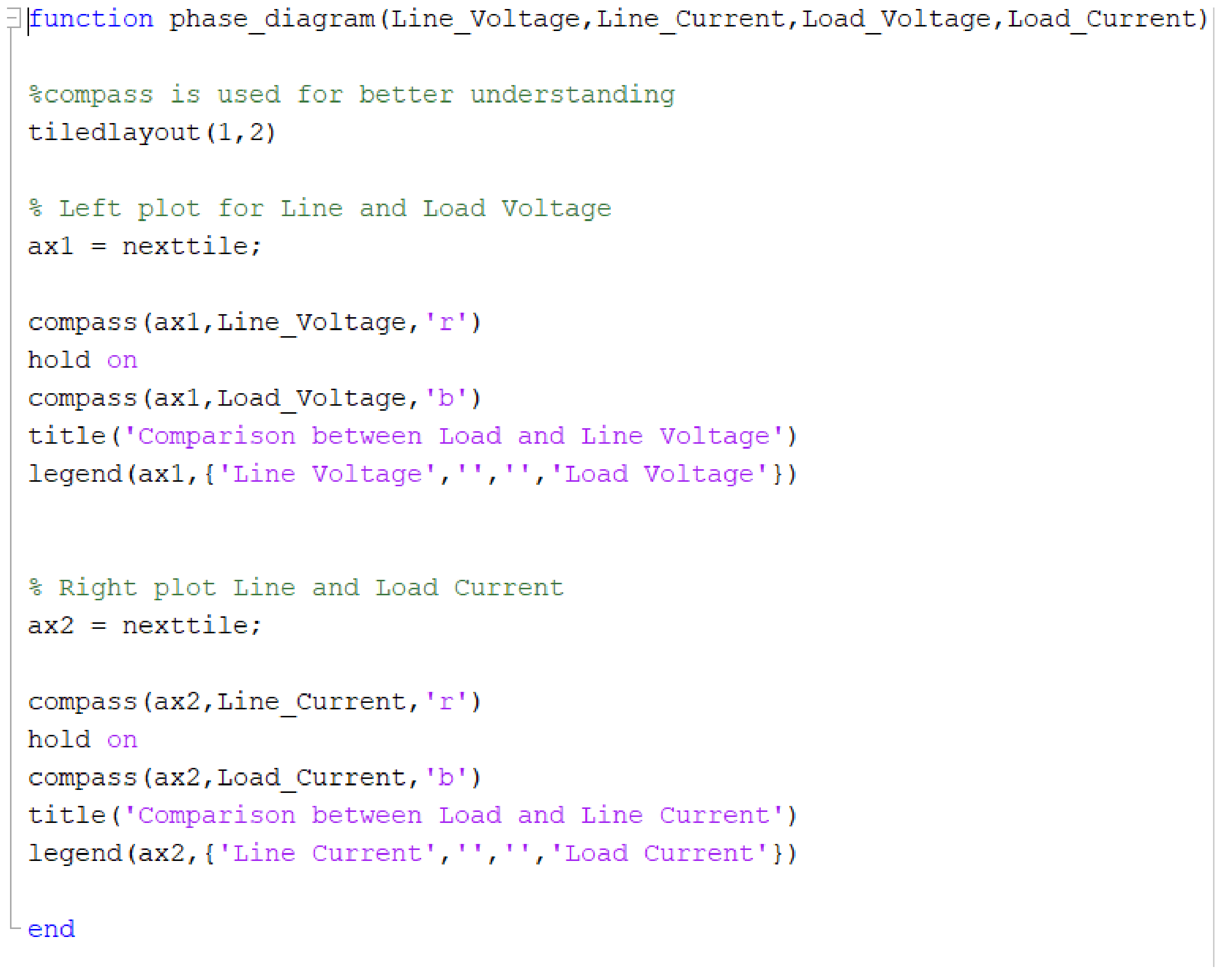
Pf = cos(theda);   %for using inverse cos(atan(…..)) was used here 'a' signifies
I   inverse in matlab

For the leading lagging part we know that if Qtotal = (-)ve than its leading

We used this info.But we didn't do this calculation in function rather we did this in the respective gui in this case deltatodelta gui.

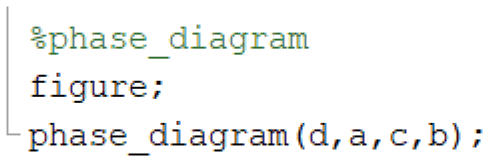
```
if( (e(1,3)) > 0)
    set(handles.param,'string','Lagging');
elseif( num2str(e(1,3)) == 0)
    set(handles.param,'string','Unity');
else
    set(handles.param,'string','Leading');
end
```

For the phase diagram part we made a function.This function was used in all the four combinations.



```
function phase_diagram(Line_Voltage,Line_Current,Load_Voltage,Load_Current)

%compass is used for better understanding
tiledlayout(1,2)

% Left plot for Line and Load Voltage
ax1 = nexttile;

compass(ax1,Line_Voltage,'r')
hold on
compass(ax1,Load_Voltage,'b')
title('Comparison between Load and Line Voltage')
legend(ax1,{'Line Voltage','','','Load Voltage'})


% Right plot Line and Load Current
ax2 = nexttile;

compass(ax2,Line_Current,'r')
hold on
compass(ax2,Load_Current,'b')
title('Comparison between Load and Line Current')
legend(ax2,{'Line Current','','','Load Current'})

end
```

The function was called in the GUI.It was called in the “SIMULATE” pussbutton section so that it could autogenerate at the time of simulation.



```
%phase_diagram
figure;
phase_diagram(d,a,c,b);
```

In this function values of line,load currents,line,load voltages are required so we called the function at the end so that the values would exist.This function uses two compass type plots to compare the relations of line,load currents,line,load voltages.On the left we have phase diagram of lineand load voltages and on the right the other two.

Same procedures were followed in other cases (Y-Y,Y-delta,delta-delta) also.But due to difference in the diagrams matrix1,matrix2 changed a bit.As a result the current_matrix also changed.As the relation changed line,load currents were gained following the diagrams.Same case also happened for line,load voltages.

For the total complex,real,reactive power,power factor,phase diagram the procedure remained same as they only need the obtained load_current matrix and given Zload_matrix.

<u>Y-Y:</u>

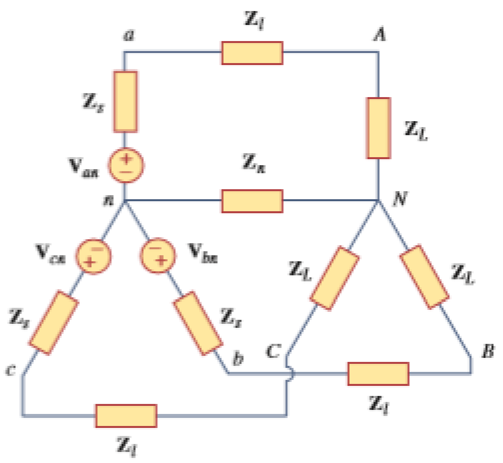


Figure 12.9
A balanced Y-Y system, showing the source, line, and load impedances.

Related mesh analysis:

```
Matrix1 = [Z_Line1 + Load1 + Load2 + Z_Line2 -Load2 - Z_Line2;...
    -Z_Line2 - Load2 Z_Line2 + Z_Line3 + Load2 + Load3];
Matrix2 = [V_Phase1 - V_Phase2;V_Phase2 - V_Phase3];
Current_Matrix = inv(Matrix1) * (Matrix2);
```

Y-del:

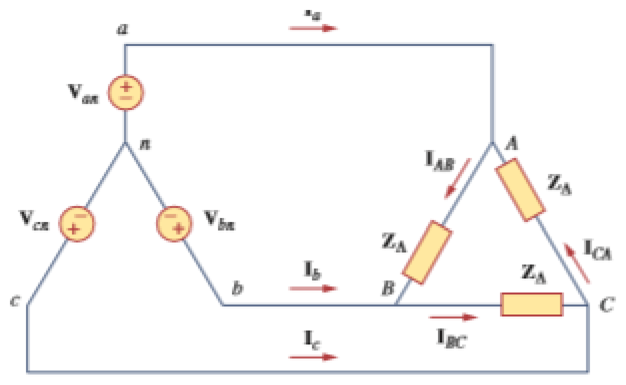


Figure 12.14
Balanced Y-Δ connection.

Related mesh analysis:

```
Matrix1 = [Z_Line1 + Load1 + Z_Line2 -Z_Line2 -Load1;...
    -Z_Line2 Z_Line2 + Load2 + Z_Line3 -Load2;...
    -Load1 -Load2 Load1 + Load2 + Load3];

Matrix2 = [V_Phase1 - V_Phase2;V_Phase2 - V_Phase3;0];

Current_Matrix = inv(Matrix1) * Matrix2;
```

Del-Y:

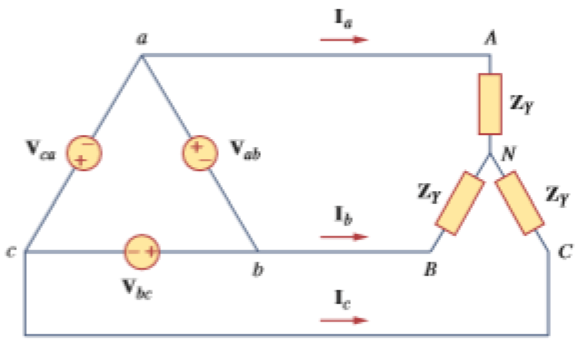


Figure 12.18
A balanced Δ-Y connection.

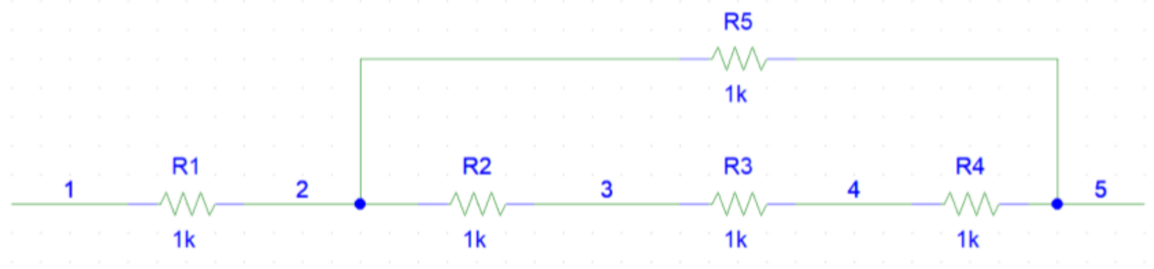
Related mesh analysis:

```
Matrix1 = [(Z_Line1 + Load1 + Load2 + Z_Line2),-(Z_Line2 + Load2);...
    -(Z_Line2 + Load2),(Z_Line2 + Load2 + Load3 + Z_Line3)];

Matrix2 = [V_Phase1;V_Phase2];

Current_Matrix = inv(Matrix1) * Matrix2;
```

<u>Bonus feature:</u>

<u>The circuit solver part --- finding equivalent impedence:</u>

In this part we solved the load and line impedences to get their equivalent impedences.The goal of this is that so that the user doesn't have to calculate equivalent impedence himself.This code takes netlist from user.User inputs data as a matrix.Lets consider the matrix is Z and starting node n1 and ending node n2.Here,Z is a nX3 matrix that contains every impedences with the nodes similar to PSpice.The first two elements of the row contains the nodes of an impedence and the third element contains the value of the impedence.The n1 and n2 is the nodes across which the user want to calculate the equivalent impedence.



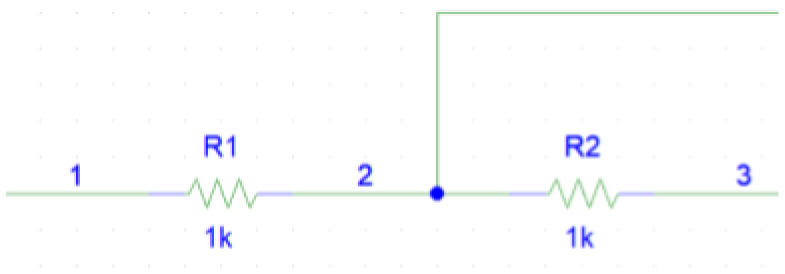
If we consider this circuit,n1 = 1, n2 = 5.We want to calculate equivalent impedence for this circuit.

The algorithm is simple.

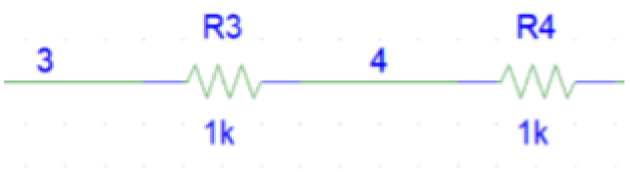
Firstly,we create a function that only works with the parallel impedences.For this purpose we take the matrix Z and search for rows that contains same nodes.If both nodes of two impedences are same than they are parallel.So,if the function finds two rows containing same nodes,it deletes one row and changes the other with equivalent impedence.The 'para' function does this job.For this it follows this eqn,

Zparallel = (Z1*Z2)/(Z1+Z2)

Another function was created for the series circuits.To find the series impedence the programme searches for rows with only one node in common.Once we find two rows that have a common node,there arises two cases.The common node can either have an extra branch in it or not.



Here,common node has extra branch so R1 and R2 not in series.



Here,common node 4 doesn't have extra branch.So,R3 and R4 in parallel.In case of extra branch 'ex_branch' does its job.For finding equivalent the formula is easy,

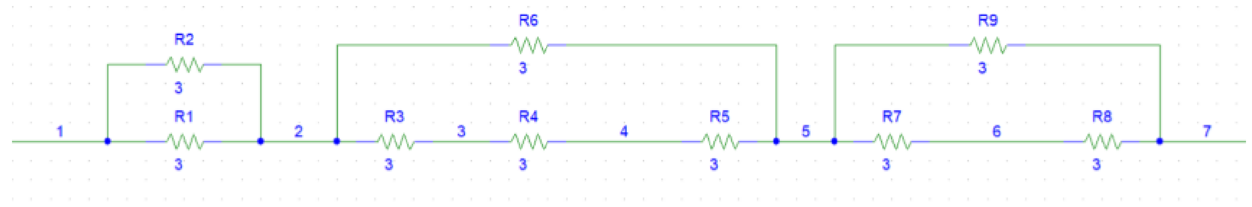
Zseries = Z1+Z2

The series function deletes one row and changes the other with proper start,end node and of course equivalent impedence.

One caution was maintained here.Avoiding of vanishing n1and n2.These are critical as they are the start and end node.If n1 or n2 are common nodes and they have extra branch the loop continues without adding them up.

We finally run a loop in 'equivalent' function after all these.First,the loop runs for parallel function and then for series function again and again.We keep checking the row numbers with the previous ones in the second loop.If we find that the row numbers is same as the previous one in the second loop,we can say that the matrix doesn't have any series connection left till then.So,we break the loop and do parallel again and again and start the second loop.This process is done until there is only one row left which contains the n1 and n2 and the equivalent impedence across the nodes.

<u>Verification of working:</u>



We checked for euivalent impedence for this circuit.Trial code for this one

```
clc;


clear all;
close all;

Z=       [1 2 3;
         1 2 3;
         2 3 3;
         3 4 3;
         4 5 3;
         2 5 3;
         5 6 3;
         6 7 3;
         5 7 3;];

Z1=equivalent(Z,1,7)
```

And the output we got:

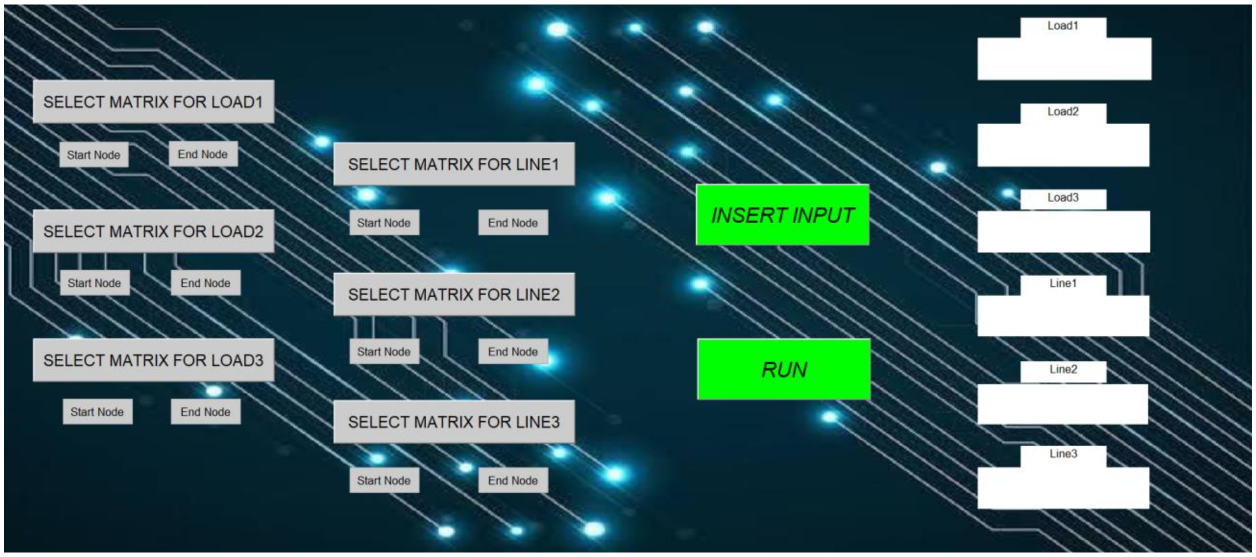
```
Z  =    1.0000    7.0000    5.7500

Z1 =    5.7500
```

Z is the equivalent matrix.n1 = 1,n2 = 7 and its value is 5.75 ohms which matches hand calculation.

<u>How to take input from user:</u>

For this case the user has to create six text file (three for line impedences,three for load impedences).This files will contain info of the impedences start node,end node and values.After creating the user will input the in the ‘GUI_Equiv_imp’ GUI.

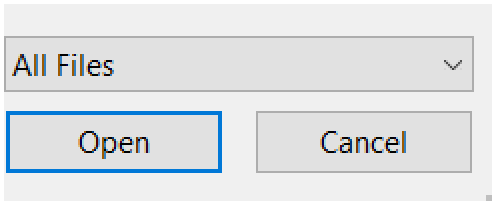
The GUI they will get by clicking ‘GUI_Equiv_imp’

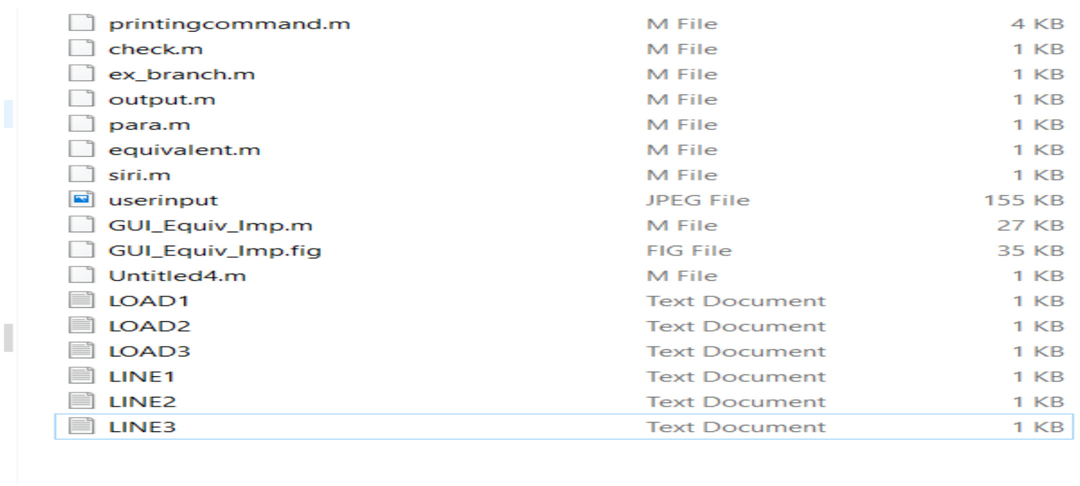


Here now they will select the text files in each six case.

They have to follow the order serially.

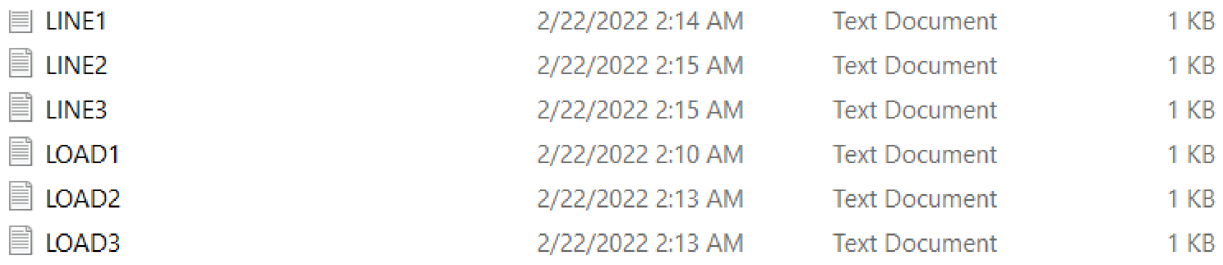
First they will click on “Select matrix for load1”.At first the text files wont be visible.User has to change matlab files to all files and the text fileswill be visible.



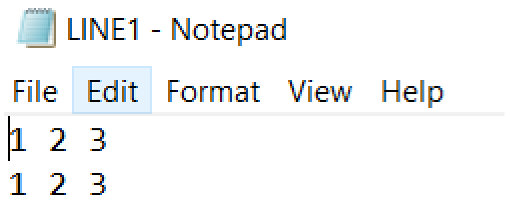


printingcommand.m	M File	4 KB
check.m	M File	1 KB
ex_branch.m	M File	1 KB
output.m	M File	1 KB
para.m	M File	1 KB
equivalent.m	M File	1 KB
siri.m	M File	1 KB
userinput	JPEG File	155 KB
GUI_Equiv_Imp.m	M File	27 KB
GUI_Equiv_Imp.fig	FIG File	35 KB
Untitled4.m	M File	1 KB
LOAD1	Text Document	1 KB
LOAD2	Text Document	1 KB
LOAD3	Text Document	1 KB
LINE1	Text Document	1 KB
LINE2	Text Document	1 KB
LINE3	Text Document	1 KB

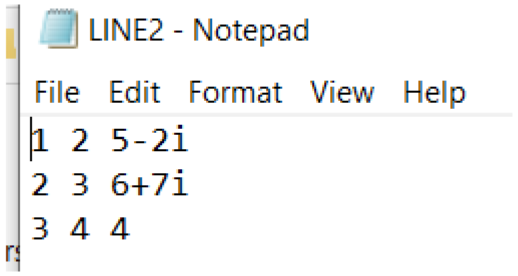
When they complete all six inputs they will run



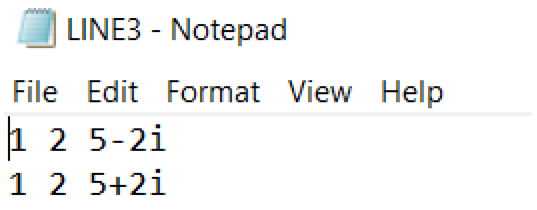
LINE1	2/22/2022 2:14 AM	Text Document	1 KB
LINE2	2/22/2022 2:15 AM	Text Document	1 KB
LINE3	2/22/2022 2:15 AM	Text Document	1 KB
LOAD1	2/22/2022 2:10 AM	Text Document	1 KB
LOAD2	2/22/2022 2:13 AM	Text Document	1 KB
LOAD3	2/22/2022 2:13 AM	Text Document	1 KB



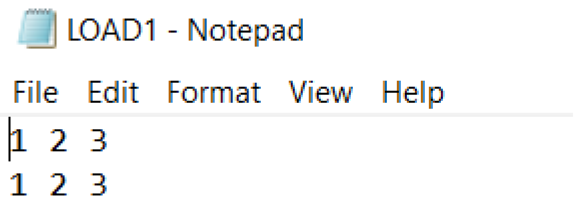
```
1 2 3
1 2 3
```



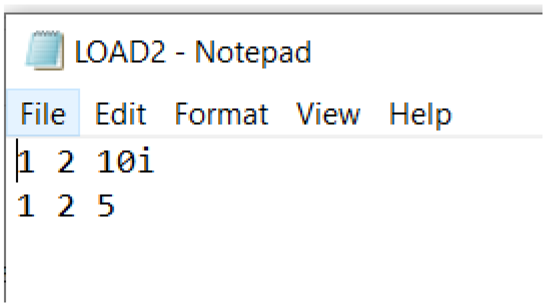
```
1 2 5-2i
2 3 6+7i
3 4 4
```



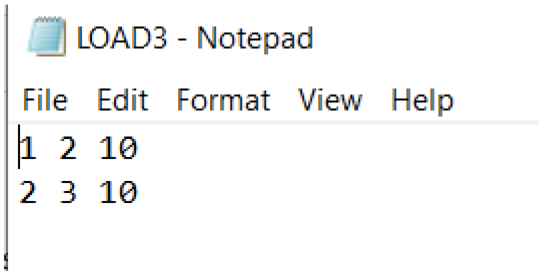
```
1 2 5-2i
1 2 5+2i
```



```
1 2 3
1 2 3
```

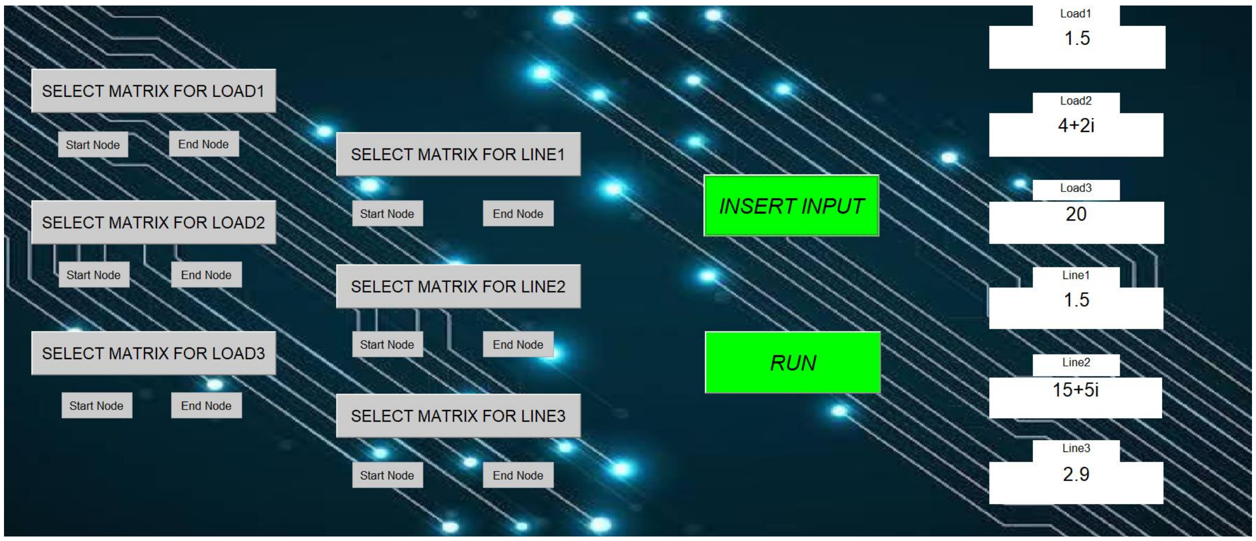


```
1 2 10i
1 2 5
```



```
1 2 10
2 3 10
```

Output:



Load1: 1.5
Load2: 4+2i
Load3: 20
Line1: 1.5
Line2: 15+5i
Line3: 2.9

After giving all six inputs user has to click “INSERT INPUT” button.Then they will get the equivalent impedence output.Needless to say all value matches theoraticsl calculation.

Problems faced in the project and how they were solved:

◊ How to plot the phase diagram plot was a big concern.Any normal axes plot didn’t make any sense.Finally after a long googling we came across compass type of plot.We implemented it and it was perfect!

◊The equivalent maker parts gui ‘GUI_Equiv_imp’ cannot take matrix as an input.This caused a great problem for us in taking matrix as input.The solution we came across is “dlmread” matlab built_in function.This function can read text file and give output as a matrix.

◊When the overall project was together under ‘GUI_Equiv_imp’ GUI there was many complexities.One of them is that data is taken input in this gui but used in other guis(Y-Y,Y-delta,…).To pass the data we used ‘setappdata’ and ‘getappdata’ built_in functions.But after we detached that equivalent finder part the problem had vanished.Taking input from user directly this has become quite easy and worth the time for detaching the ‘GUI_Equiv_imp’ GUI.Now the four combination GUI are under ‘menu.m’ gui.

These were the major problems we faced,but we faced countless other problems too.If we explain each one the report will be very lengthy.Those were solved using trial and error,googling,youtubing,discussion and what not!

Possible application of the project:

This project can solve three-phase circuits accurately.This can be used in the power supply tasks to calculate and manage supplying power.As for power supplying the supplyer has to know how much output is needed in the load side.So,he can set the voltages accordingly.As three-phase power is used commonly this project has big scope.

Accuracy of project:

Its safe to say this project gave accurate outputs.The deviations that occurred is due to conversion from R<theda to A+iB form of source voltages.If we had put the input accurately the output would have matched(but putting accurately after infinite decimal points is impossible needless to say).But that deviation is so meagre that it can practically be discarded.

Conclusion:

The goal of the project was to create a fully functioning three-phase circuit analyser.This project can do that very well.

If a complex impedence configuration occurs then user has to calculate its equivalent impedence.To ease their work,rquivalent finding GUI and functions were created.

If the user doesn't feel comfortable than there is hardly any point in making a project.So,when taking input from 'GUI_Equiv_imp' was found really troublesome it was discarded altogether and made a separate file making the 'menu' GUI the main GUI which is very user friendly.

So,we can say that this project was a success.

And of course,some further improvements can be done like reducing the already negligible error to zero.

More Examples:

Ex-1:

This example was done using the equivalent gui implemented in the main code(before the separation)

Calculate the line currents in the three-wire Y-Y system of Fig. 12.13. **Example 12.2**

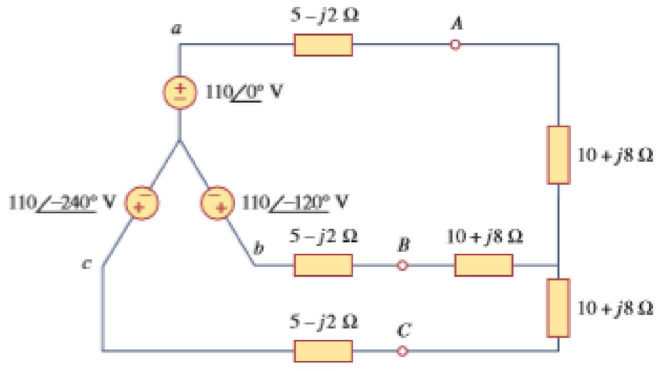
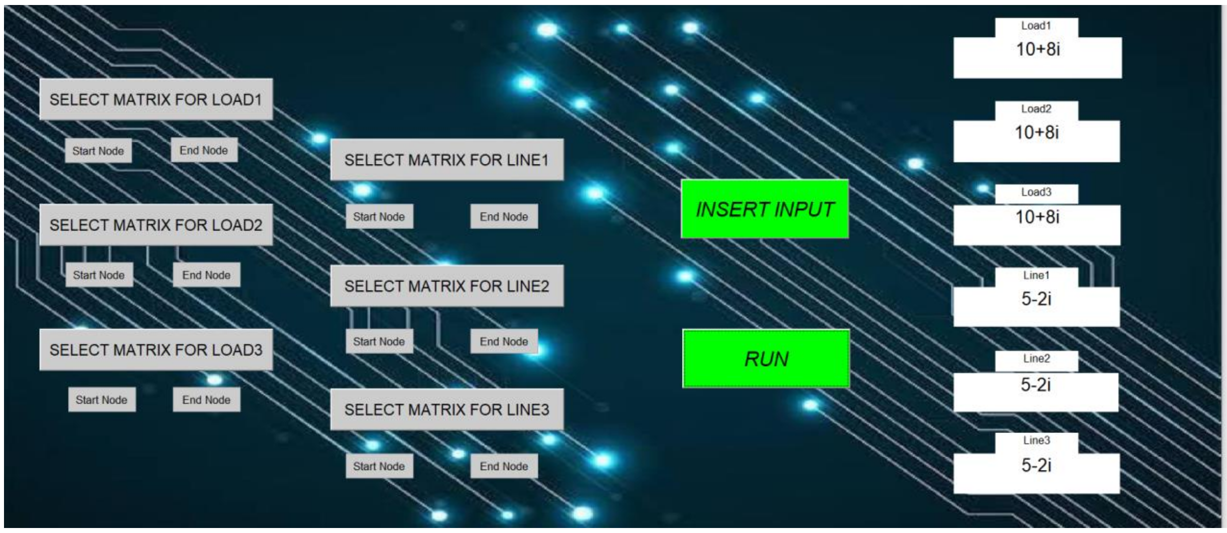
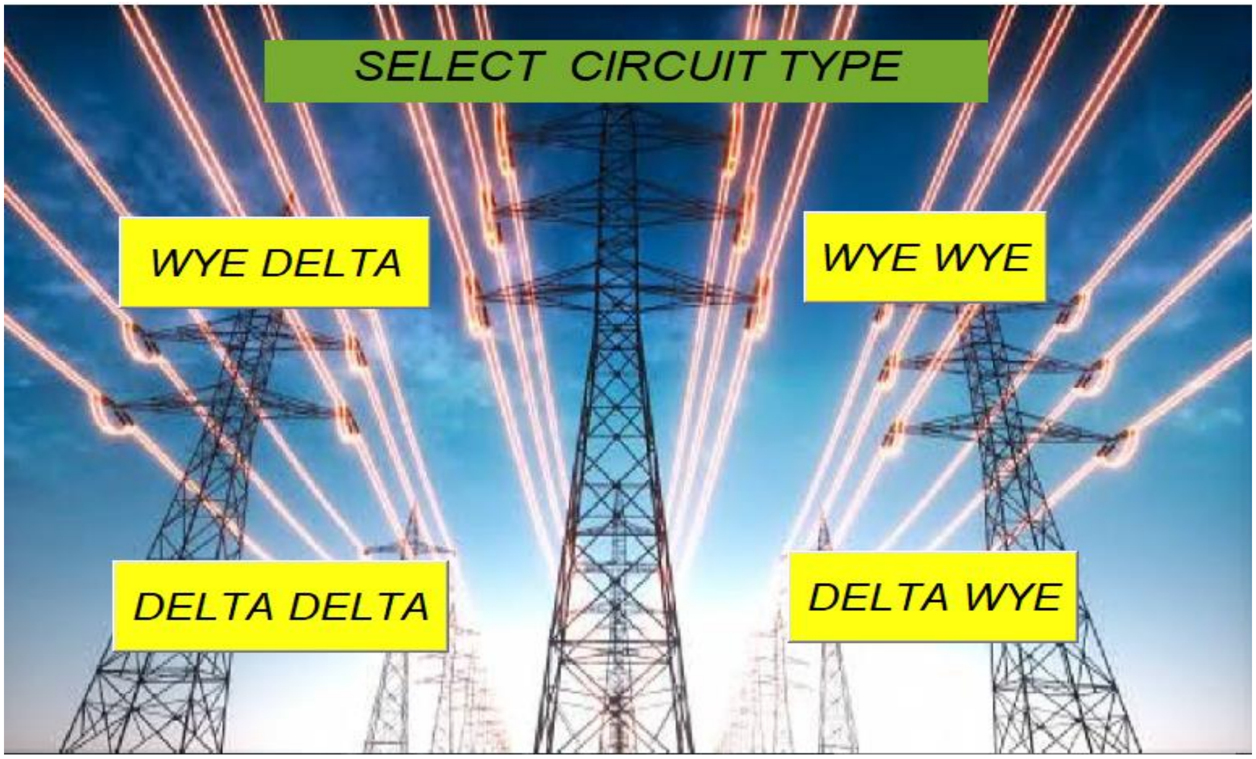
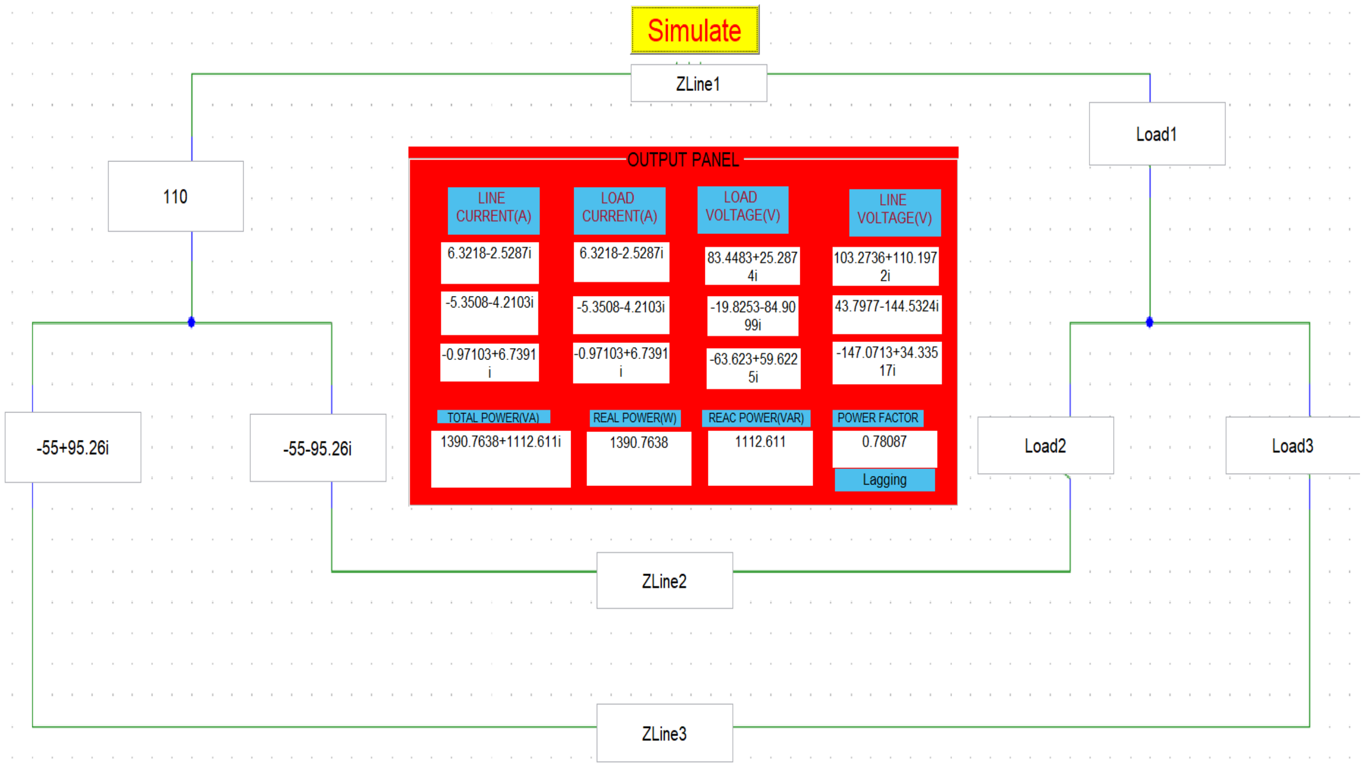


Figure 12.13
Three-wire Y-Y system; for Example 12.2.



Load1	10+8i
Load2	10+8i
Load3	10+8i
Line1	5-2i
Line2	5-2i
Line3	5-2i





OUTPUT PANEL

LINE CURRENT(A)	LOAD CURRENT(A)	LOAD VOLTAGE(V)	LINE VOLTAGE(V)
6.3218-2.5287i	6.3218-2.5287i	83.4483+25.2874i	103.2736+110.1972i
-5.3508-4.2103i	-5.3508-4.2103i	-19.8253-84.9099i	43.7977-144.5324i
-0.97103+6.7391i	-0.97103+6.7391i	-63.623+59.6225i	-147.0713+34.33517i

TOTAL POWER(VA)	REAL POWER(W)	REAC POWER(VAR)	POWER FACTOR
1390.7638+1112.611i	1390.7638	1112.611	0.78087
			Lagging

Calculate the line currents in the three-wire Y-Y system of Fig. 12.13. **Example 12.2**

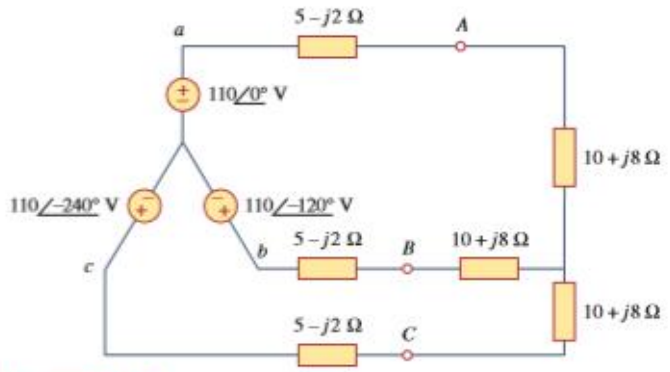


Figure 12.13
Three-wire Y-Y system; for Example 12.2.

Solution:
The three-phase circuit in Fig. 12.13 is balanced; we may replace it with its single-phase equivalent circuit such as in Fig. 12.12. We obtain $\mathbf{I}_a$ from the single-phase analysis as

$$\mathbf{I}_a = \frac{\mathbf{V}_{an}}{\mathbf{Z}_Y}$$

where $\mathbf{Z}_Y = (5 - j2) + (10 + j8) = 15 + j6 = 16.155\underline{/21.8^\circ}$. Hence,

$$\mathbf{I}_a = \frac{110\underline{/0^\circ}}{16.155\underline{/21.8^\circ}} = 6.81\underline{/-21.8^\circ}\text{ A}$$

Since the source voltages in Fig. 12.13 are in positive sequence, the line currents are also in positive sequence:

$$\mathbf{I}_b = \mathbf{I}_a\underline{/-120^\circ} = 6.81\underline{/-141.8^\circ}\text{ A}$$
$$\mathbf{I}_c = \mathbf{I}_a\underline{/-240^\circ} = 6.81\underline{/-261.8^\circ}\text{ A} = 6.81\underline{/98.2^\circ}\text{ A}$$

Ia = 6.81<-21.8 =6.3229-2.5291i (using calculator)

Output for respective line current = 6.3218-2.5287i

So its safe to say output provided by the code matches theoratical calculation.The error that occurred is due to conversion from R<theda to A+iB form of source voltages.

No,phase diagram is shown as this example was taken before separating equivalent circuit finder.That feature was added after the separation.

Ex-2: (after the separation with equivalent finding GUI)

Unbalanced Y-Y:

Example 12.10

For the unbalanced circuit in Fig. 12.25, find: (a) the line currents, (b) the total complex power absorbed by the load, and (c) the total complex power absorbed by the source.

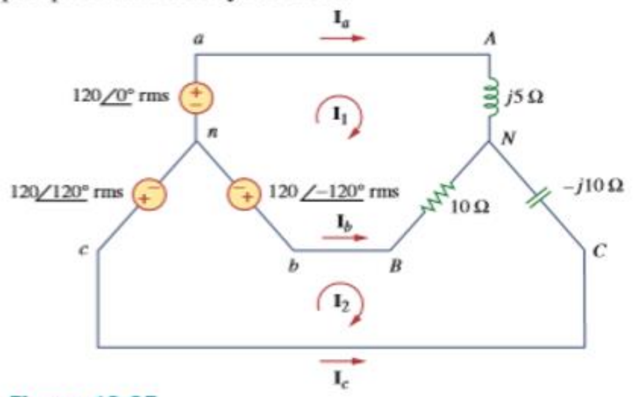
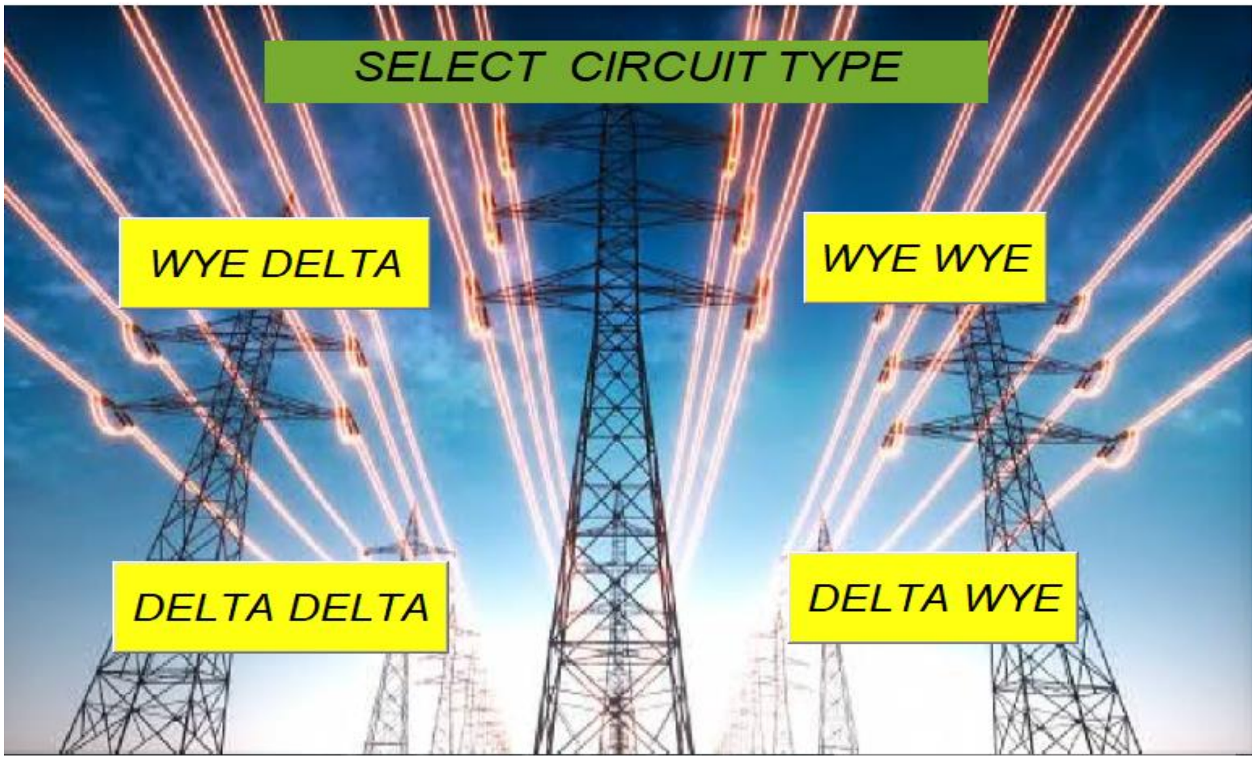
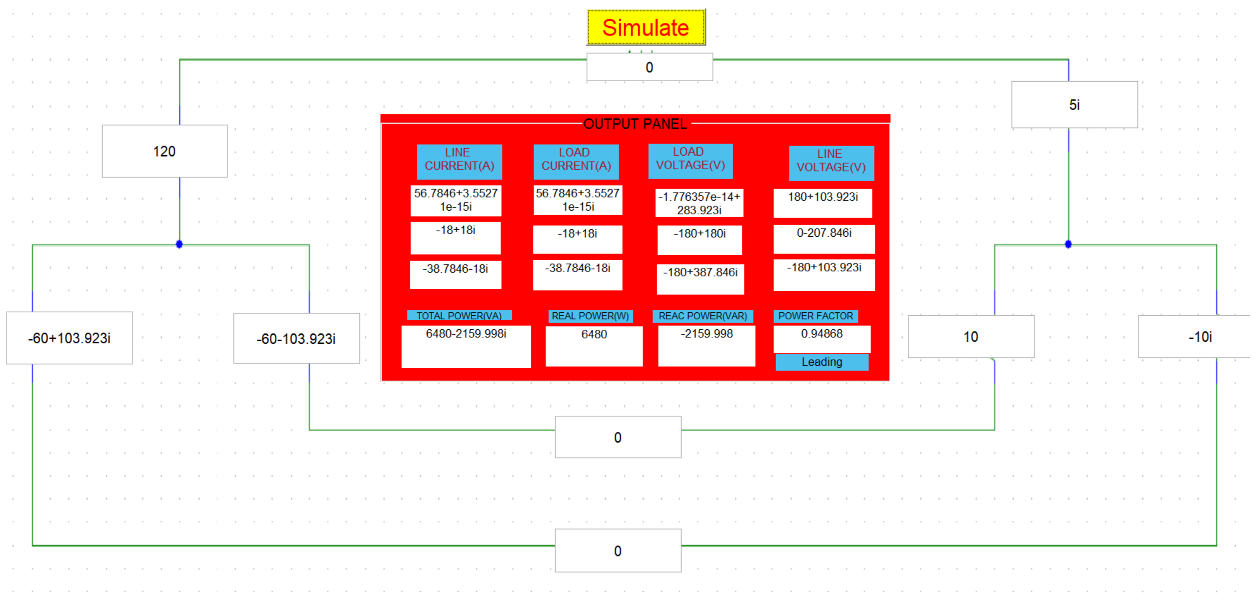
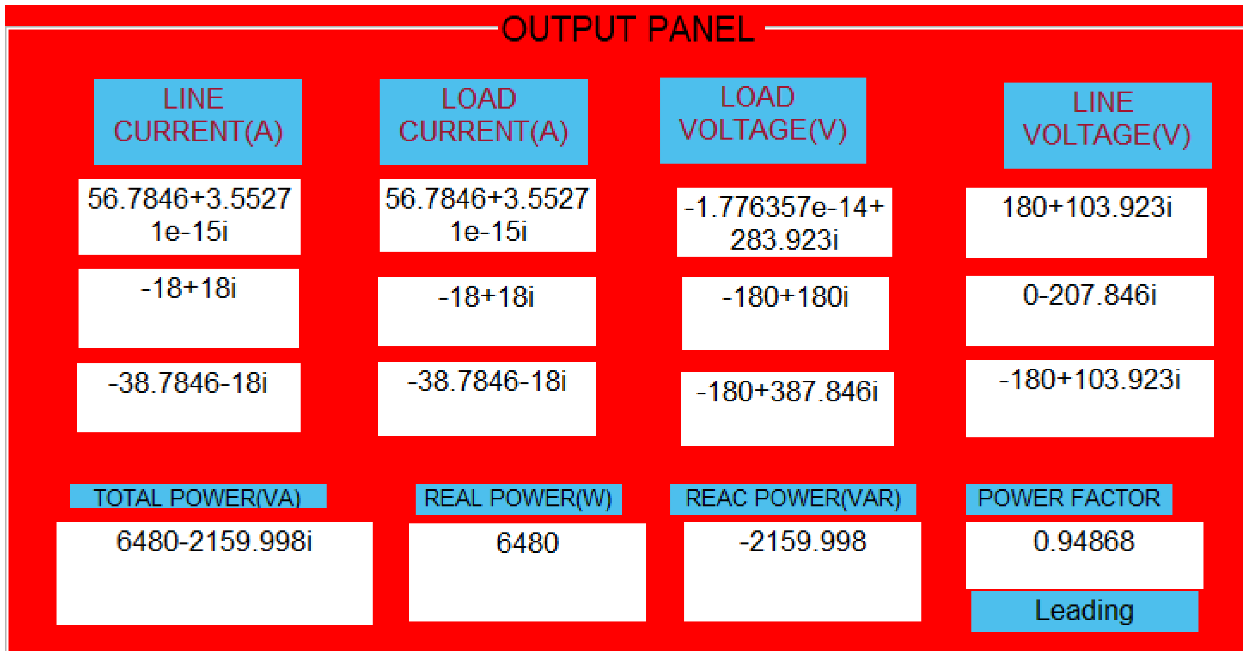


Figure 12.25
For Example 12.10.

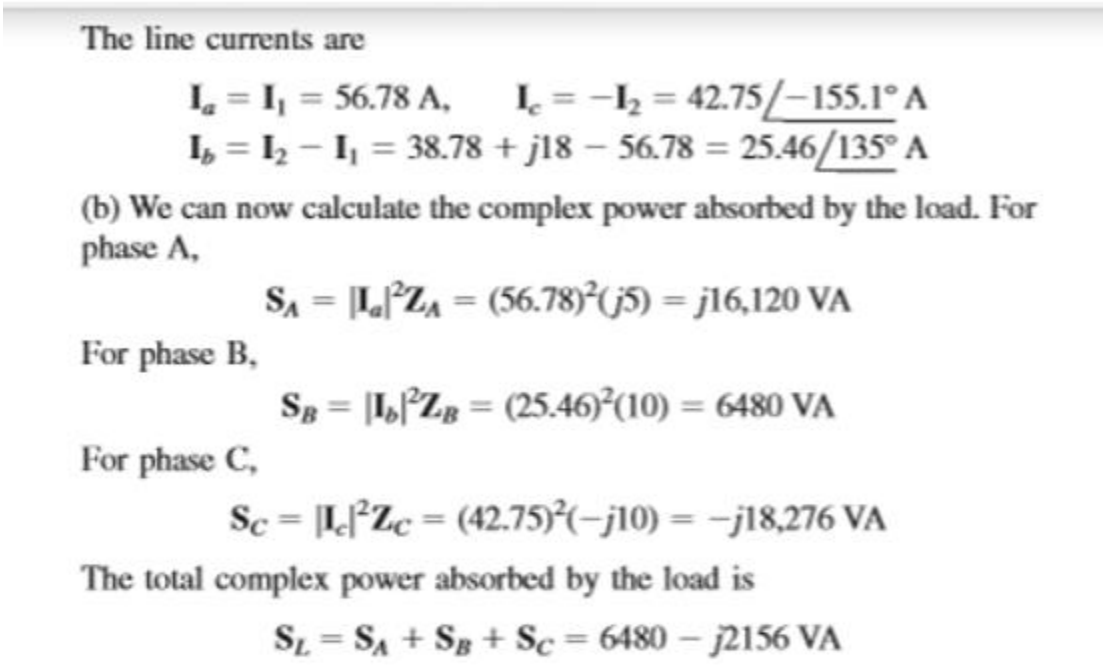





OUTPUT PANEL

LINE CURRENT(A)	LOAD CURRENT(A)	LOAD VOLTAGE(V)	LINE VOLTAGE(V)
56.7846+3.5527 1e-15i	56.7846+3.5527 1e-15i	-1.776357e-14+ 283.923i	180+103.923i
-18+18i	-18+18i	-180+180i	0-207.846i
-38.7846-18i	-38.7846-18i	-180+387.846i	-180+103.923i

TOTAL POWER(VA)	REAL POWER(W)	REAC POWER(VAR)	POWER FACTOR
6480-2159.998i	6480	-2159.998	0.94868
			Leading



The line currents are

$$\mathbf{I}_a = \mathbf{I}_1 = 56.78 \text{ A}, \qquad \mathbf{I}_c = -\mathbf{I}_2 = 42.75\underline{/-155.1^\circ} \text{ A}$$

$$\mathbf{I}_b = \mathbf{I}_2 - \mathbf{I}_1 = 38.78 + j18 - 56.78 = 25.46\underline{/135^\circ} \text{ A}$$

(b) We can now calculate the complex power absorbed by the load. For phase A,

$$\mathbf{S}_A = |\mathbf{I}_a|^2\mathbf{Z}_A = (56.78)^2(j5) = j16{,}120 \text{ VA}$$

For phase B,

$$\mathbf{S}_B = |\mathbf{I}_b|^2\mathbf{Z}_B = (25.46)^2(10) = 6480 \text{ VA}$$

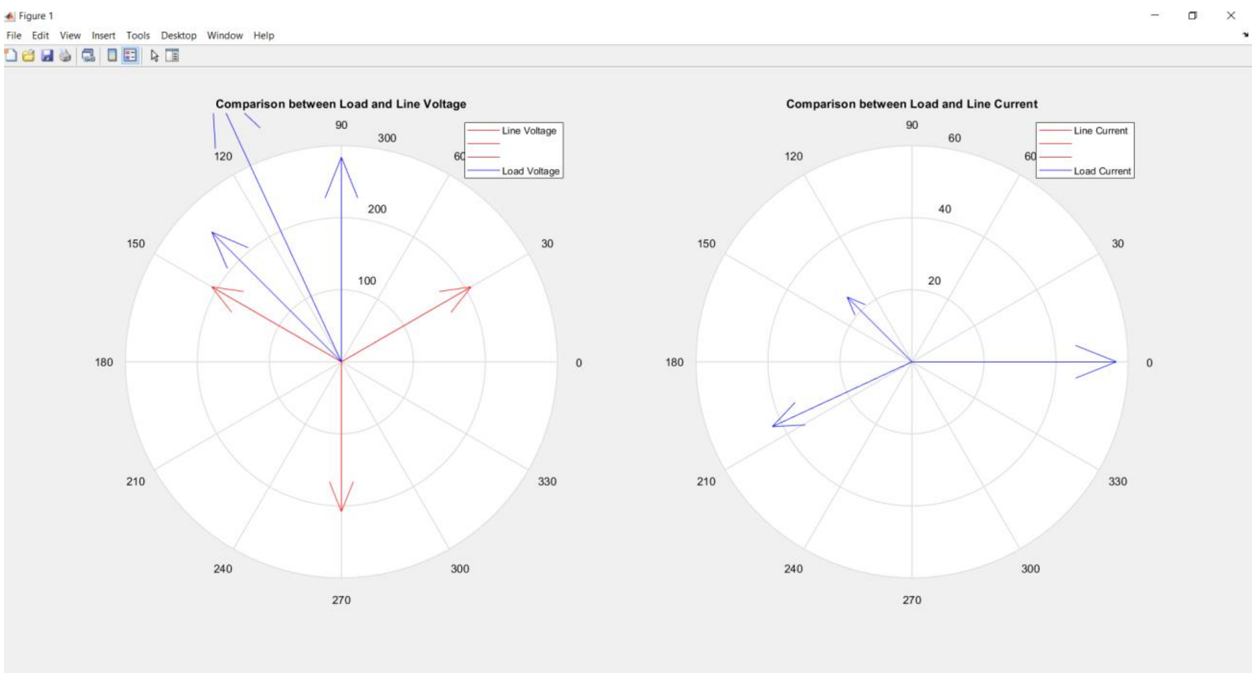
For phase C,

$$\mathbf{S}_C = |\mathbf{I}_c|^2\mathbf{Z}_C = (42.75)^2(-j10) = -j18{,}276 \text{ VA}$$

The total complex power absorbed by the load is

$$\mathbf{S}_L = \mathbf{S}_A + \mathbf{S}_B + \mathbf{S}_C = 6480 - j2156 \text{ VA}$$

The results of the output panel matches with sadiku.The negligible difference that occurred is due to using A+iB form instead of R<theda form.



The phase diagram added after detaching the equivalent_gui.Gives us a perspective about the phases between line and load voltages,line and load currents.

Link for github:

https://github.com/sami3033/Three-phase-circuit-solver