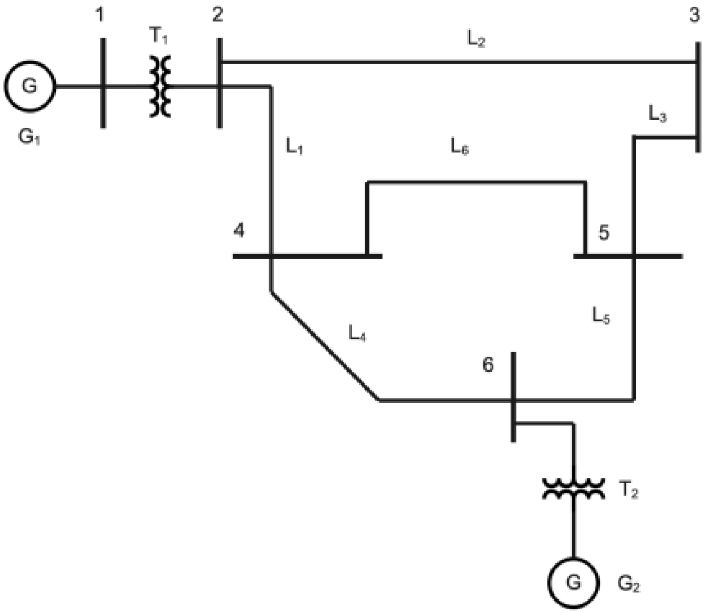
Jack Carnovale & Aiden Landis

Simulator Milestone 1 Verification

For this milestone, our simulator was built to be able to add lines and transformers to a system and create the associated YBus Matrix. See the documentation to learn about how this software works. This document will outline a verification of results from the simulator as compared to results from MATLAB and PowerWorld.

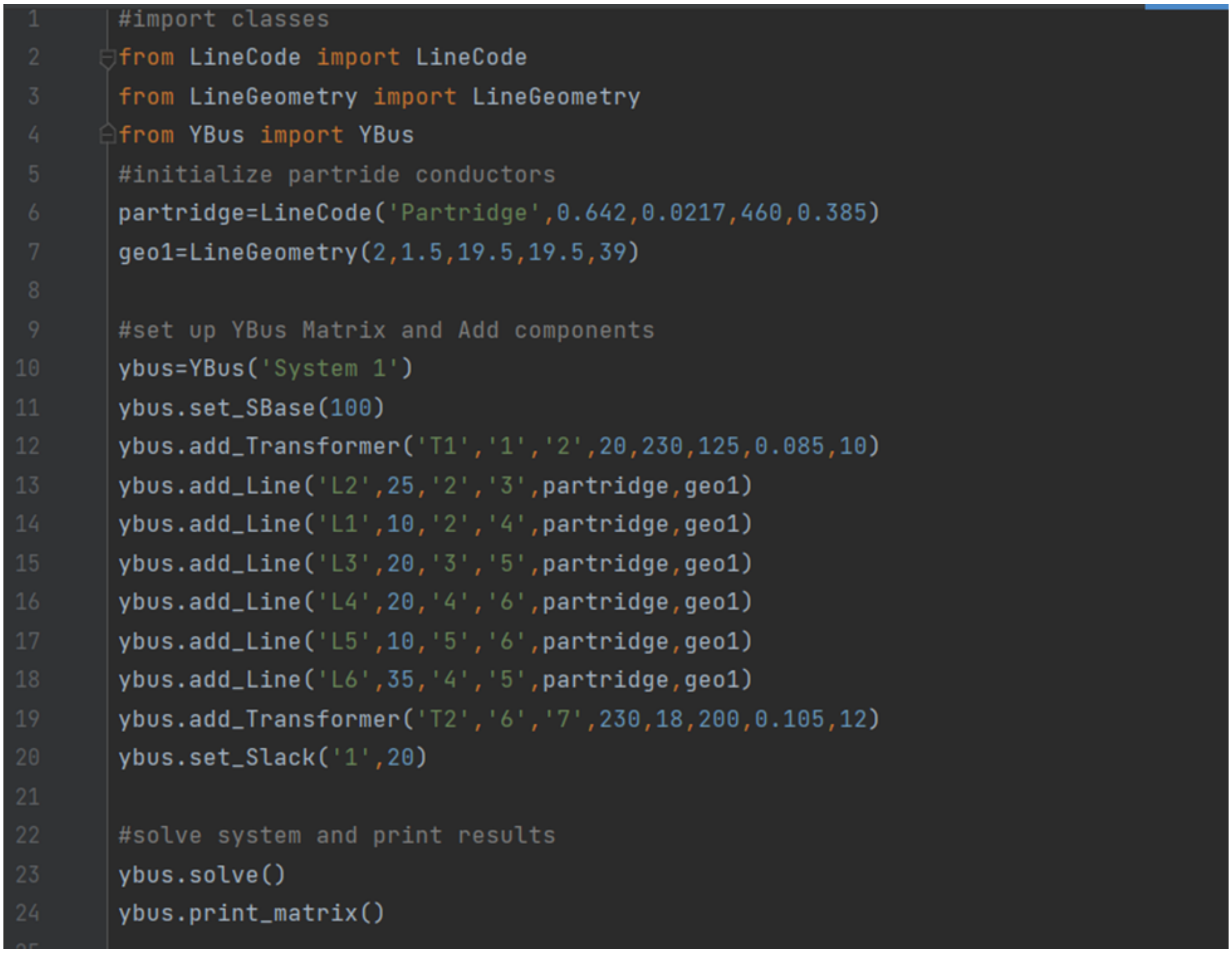
6-Node Looped Transmission System



Python Results:

First, the code was run using the system given in the assignment document that is shown above. The main function below was used to create and test the system. Notice that only raw data was input into the YBus class and that all calculation of per unit impedance and admittance values was done internally. The YBus Matrix is only solved once the .solve() function is called and results can be seen by calling the .print_matrix() function. See the main code and the associated output below.

Main Code:

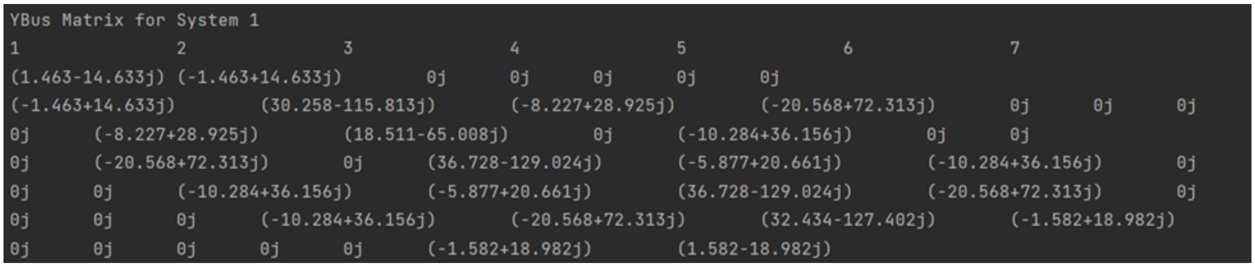


```python
#import classes
from LineCode import LineCode
from LineGeometry import LineGeometry
from YBus import YBus
#initialize partride conductors
partridge=LineCode('Partridge',0.642,0.0217,460,0.385)
geo1=LineGeometry(2,1.5,19.5,19.5,39)

#set up YBus Matrix and Add components
ybus=YBus('System 1')
ybus.set_SBase(100)
ybus.add_Transformer('T1','1','2',20,230,125,0.085,10)
ybus.add_Line('L2',25,'2','3',partridge,geo1)
ybus.add_Line('L1',10,'2','4',partridge,geo1)
ybus.add_Line('L3',20,'3','5',partridge,geo1)
ybus.add_Line('L4',20,'4','6',partridge,geo1)
ybus.add_Line('L5',10,'5','6',partridge,geo1)
ybus.add_Line('L6',35,'4','5',partridge,geo1)
ybus.add_Transformer('T2','6','7',230,18,200,0.105,12)
ybus.set_Slack('1',20)

#solve system and print results
ybus.solve()
ybus.print_matrix()
```

Output:

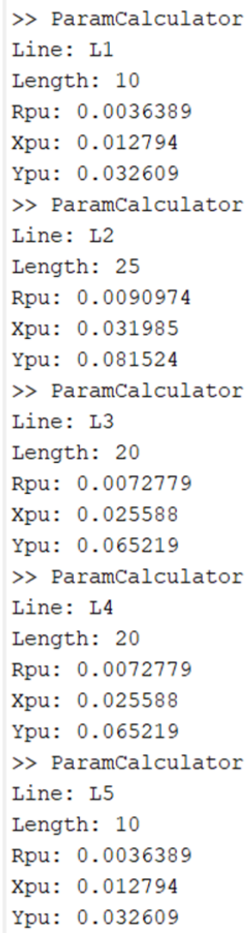


```
YBus Matrix for System 1
1               2               3               4               5               6               7
(1.463-14.633j) (-1.463+14.633j)        0j      0j      0j      0j      0j
(-1.463+14.633j)        (30.258-115.813j)       (-8.227+28.925j)        (-20.568+72.313j)       0j      0j      0j
0j      (-8.227+28.925j)        (18.511-65.008j)        0j      (-10.284+36.156j)       0j      0j
0j      (-20.568+72.313j)       0j      (36.728-129.024j)       (-5.877+20.661j)        (-10.284+36.156j)       0j
0j      0j      (-10.284+36.156j)       (-5.877+20.661j)        (36.728-129.024j)       (-20.568+72.313j)       0j
0j      0j      0j      (-10.284+36.156j)       (-20.568+72.313j)       (32.434-127.402j)       (-1.582+18.982j)
0j      0j      0j      0j      0j      (-1.582+18.982j)        (1.582-18.982j)
```

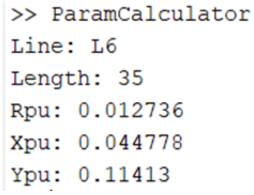
The formatting is slightly nonuniform but it clearly shows a 7x7 YBus matrix for this system with 7 buses. It is symmetrical as well, which is a requirement of the YBus matrix.

Verification:

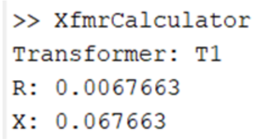
To verify these results, a few steps were taken. First, using a MATLAB script, all per unit impedances and admittances were calculated for the lines and transformers in this system. See the MATLAB results below.



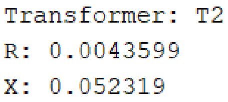
```
>> ParamCalculator
Line: L1
Length: 10
Rpu: 0.0036389
Xpu: 0.012794
Ypu: 0.032609
>> ParamCalculator
Line: L2
Length: 25
Rpu: 0.0090974
Xpu: 0.031985
Ypu: 0.081524
>> ParamCalculator
Line: L3
Length: 20
Rpu: 0.0072779
Xpu: 0.025588
Ypu: 0.065219
>> ParamCalculator
Line: L4
Length: 20
Rpu: 0.0072779
Xpu: 0.025588
Ypu: 0.065219
>> ParamCalculator
Line: L5
Length: 10
Rpu: 0.0036389
Xpu: 0.012794
Ypu: 0.032609
```



```
>> ParamCalculator
Line: L6
Length: 35
Rpu: 0.012736
Xpu: 0.044778
Ypu: 0.11413
```

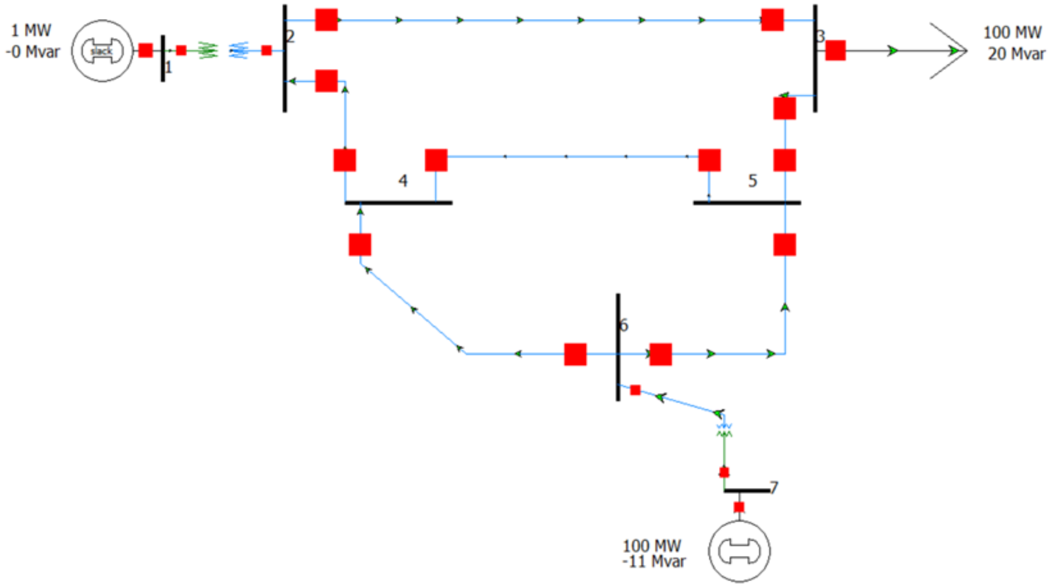


```
>> XfmrCalculator
Transformer: T1
R: 0.0067663
X: 0.067663
```

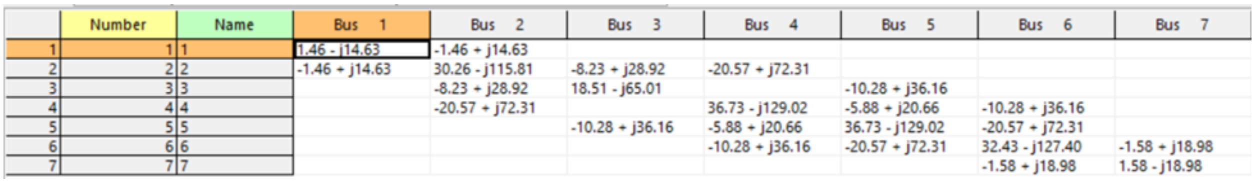


```
Transformer: T2
R: 0.0043599
X: 0.052319
```

Next, the system was created as a one-line in PowerWorld. For this simulation, the per unit values calculated above were used to define all lines and transformers in the system. See the one-line below.



Last, in PowerWorld, the YBus matrix was compared to the one calculated in the python simulator.



	Number	Name	Bus 1	Bus 2	Bus 3	Bus 4	Bus 5	Bus 6	Bus 7
1	1	1	1.46 - j14.63	-1.46 + j14.63					
2	2	2	-1.46 + j14.63	30.26 - j115.81	-8.23 + j28.92	-20.57 + j72.31			
3	3	3		-8.23 + j28.92	18.51 - j65.01		-10.28 + j36.16		
4	4	4		-20.57 + j72.31		36.73 - j129.02	-5.88 + j20.66	-10.28 + j36.16	
5	5	5			-10.28 + j36.16	-5.88 + j20.66	36.73 - j129.02	-20.57 + j72.31	
6	6	6				-10.28 + j36.16	-20.57 + j72.31	32.43 - j127.40	-1.58 + j18.98
7	7	7						-1.58 + j18.98	1.58 - j18.98

Note that all values matched for the python and PowerWorld simulators when rounded to the second decimal place. This confirmed that the python simulator could successfully create a YBus matrix for a transmission system that accurately represents the power system.