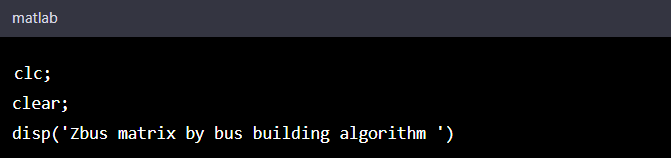
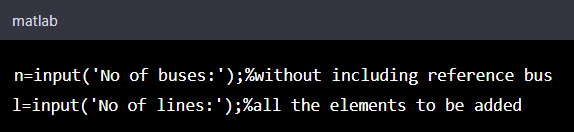
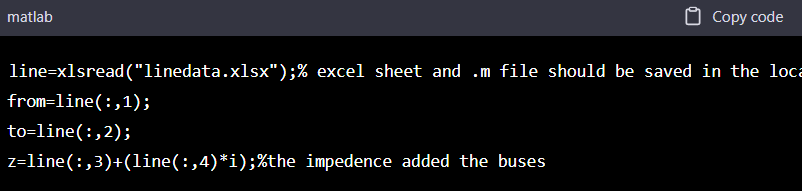
**ZBUS BUS BUILDING ALGORITHM:**



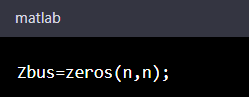
* The clc and clear commands clear the command window and the workspace, respectively. disp command is used to display a message in the command window.



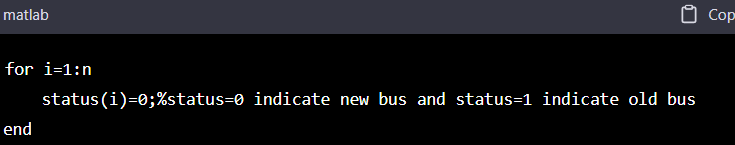
* Input function is used to take input from the user. The number of buses and the number of lines are stored in n and l variables, respectively.



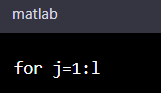
* xlsread function reads data from an Excel spreadsheet file named "linedata.xlsx" and stores it in line matrix. The first column, second column, and third and fourth columns of the line matrix are assigned to from, to, and z variables respectively. The values in the third and fourth columns of line matrix represent the resistance and reactance of the transmission line respectively. These values are combined to create a complex impedance and stored in z variable.



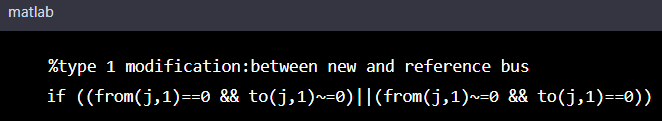
* This line creates an n x n matrix Zbus with all elements initialized to zero.



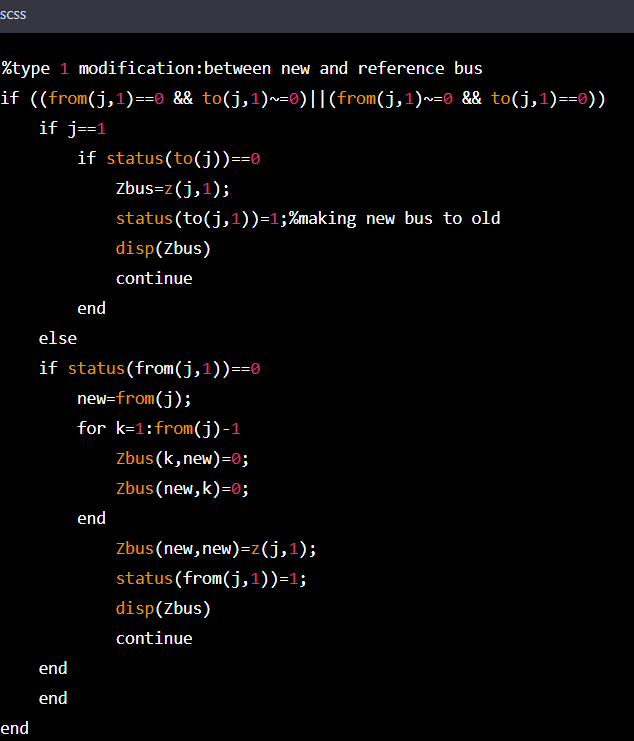
* This loop initializes a status vector with n elements to zero. The status vector is used to keep track of whether a bus is new or old. A value of 0 indicates a new bus, and a value of 1 indicates an old bus.



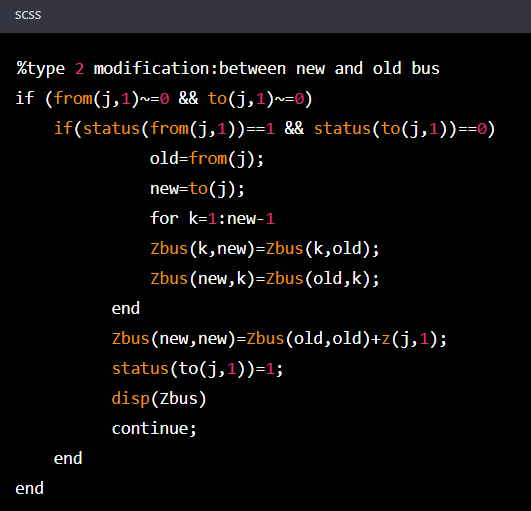
* This loop iterates l times, where l is the number of transmission lines.



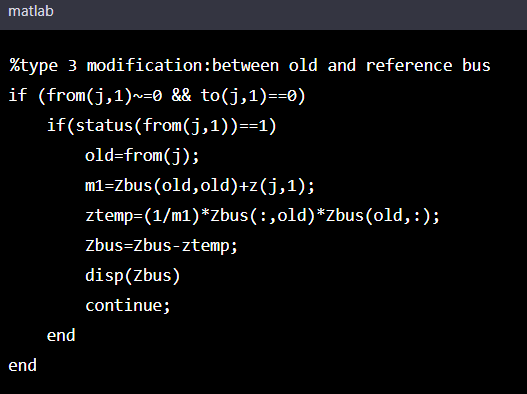
* This if statement checks if the current transmission line is between a new bus and the reference bus (i.e., the bus with ID 0). If so, it proceeds with the following steps:



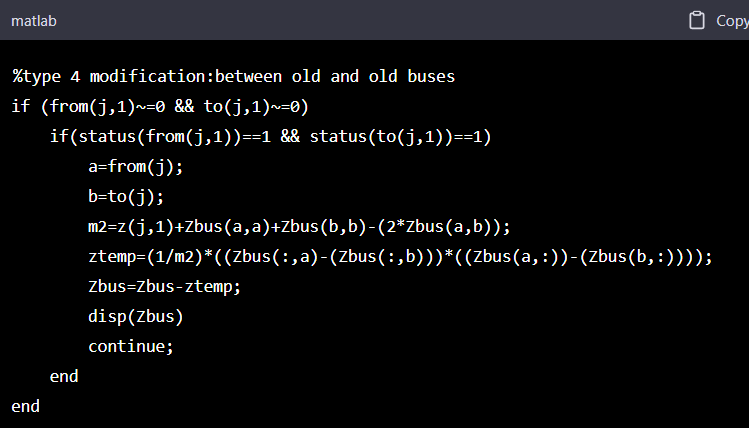
* This is the code for a type 1 modification, which occurs when a line connects a new bus to the reference bus (bus 1) or vice versa. The if statement checks if the line connects a new bus to the reference bus.
* If it does, the code checks if it is the first line being added to the Zbus matrix (j==1). If it is, then the impedances of the line are simply added to the Zbus matrix, and the status of the new bus is changed to 1 (i.e., it is marked as an old bus). If it is not the first line, then the code checks if the new bus has already been added to the Zbus matrix.
* If it has not, then a new row and column are added to the Zbus matrix for the new bus, and the impedances of the line are added to the diagonal element of the new row and column. The status of the new bus is changed to 1, and the modified Zbus matrix is printed.



* This is the code for a type 2 modification, which occurs when a line connects a new bus to an old bus.
* The if statement checks if the line connects a new bus to an old bus. If it does, the code checks if the old bus has already been added to the Zbus matrix and if the new bus has not.
* If both conditions are true, then a new row and column are added to the Zbus matrix for the new bus, and the values from the row and column of the old bus are copied into the corresponding positions in the row and column of the new bus.
* The diagonal element of the new row and column is set to the sum of the diagonal element of the old row and column and the impedance of the line. The status of the new bus is changed to 1, and the modified Zbus matrix is printed.



* The code checks if the **from** bus is an old bus (i.e., **status(from(j,1)) == 1**) and if the **to** bus is the reference bus (i.e., **to(j,1) == 0**).
* If the condition is satisfied, the code proceeds to calculate the equivalent impedance (**m1**) of the new line and the impedance of the old bus (**Zbus(old,old)**).
* Then the code computes the temporary impedance matrix (**ztemp**) using the equation: **ztemp=(1/m1)\*Zbus(:,old)\*Zbus(old,:)**.
* Finally, the code updates the impedance matrix **Zbus** by subtracting the temporary impedance matrix (**ztemp**) from **Zbus**.



* The code checks if both the **from** and **to** buses are old buses (i.e., **status(from(j,1)) == 1** and **status(to(j,1)) == 1**).
* If the condition is satisfied, the code proceeds to calculate the equivalent impedance (**m2**) of the new line and the impedances of the two old buses (**Zbus(a,a)** and **Zbus(b,b)**).
* Then the code computes the temporary impedance matrix (**ztemp**) using the equation: **ztemp=(1/m2)\*((Zbus(:,a)-(Zbus(:,b)))\*((Zbus(a,:))-(Zbus(b,:))))**.
* Finally, the code updates the impedance matrix **Zbus** by subtracting the temporary impedance matrix (**ztemp**) from **Zbus**.