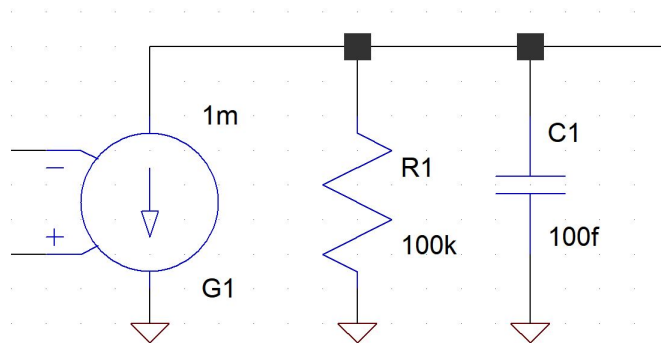# LTspice Basic Simulation Exercises
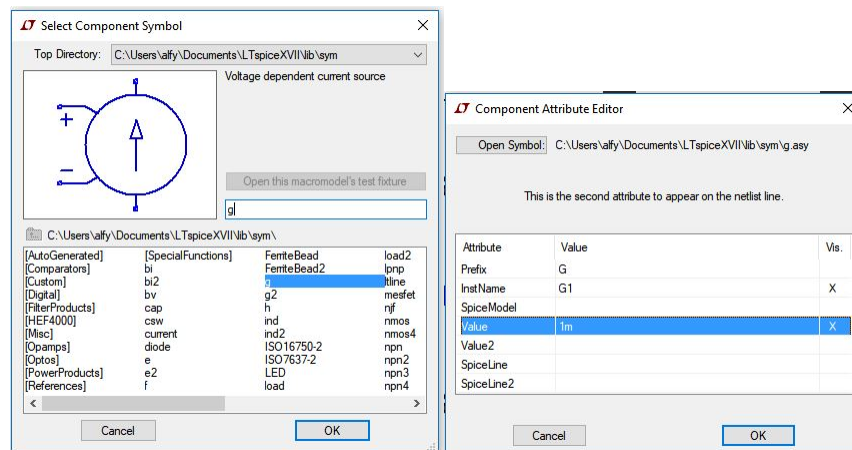
Alfred Festus Davidson
alfy7.github.io

December 4, 2017
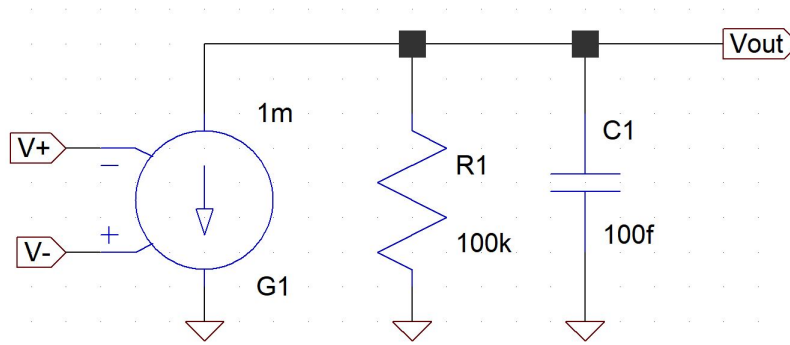
Single Stage Opamp

- Create a new schematic and draw the following circuit.

- Use the *Add Component* icon and search for *g* to add the voltage controlled current source. *g2* is also a voltage controlled current source with the input signs interchanged. Right click it and set its value to $1mS$.


- The schematic we drew is the model for a single stage opamp. If we were to use it multiple times in a schematic, it would be messy and difficult to interpret. Hence, we create a model and use the usual opamp symbol to describe the functionality of this entire circuit.

- First we label the nodes. When labeling the nodes, we can select the port type to *input* or *output*, to get a different label style. Note the labeling of *V+* and *V-* for proper sign of *Vout*.

Save this schematic as *single_stage_opamp.asc*

- Click *File->New Symbol.* We can use the default opamp symbol LTspice uses and modify it, or we can create a new one. Let us make the symbol from scratch. Draw the following symbol. Lines can be drawn by right-clicking anywhere in the screen, and selecting *Draw->Line* or using the shortcut key *L*.
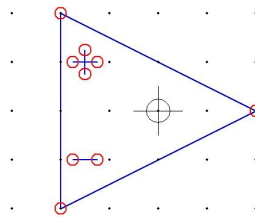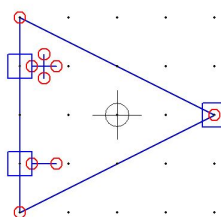

- Now we need to mark the points where connections will be made. Right-click anywhere in the screen, and select *Add Pin*
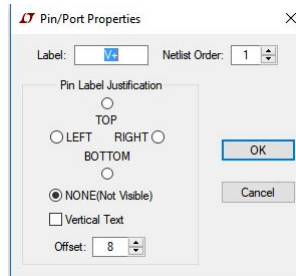

The label should have the same name as the corresponding node in the schematic. Click on the symbol to make that point a connecting pin.
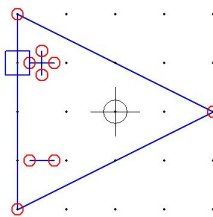

Add the 3 pins, *V+*, *V-* and *Vout*

- Right-click anywhere in the screen and choose *Attributes->Edit Attributes.* We can add a description and modify the prefix with which every instance of this opamp will be named (like R1 and R2 for resistors). If no prefix is specified, the default is *X*.

.
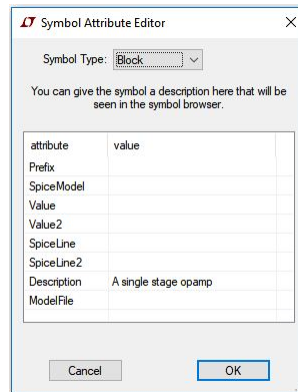
- When we have multiple opamps, LTspice will number them according to the prefix. We would also want the name to be on the schematic near the symbol. To do this, right-click anywhere in the screen and choose *Attributes->Attribute Window*
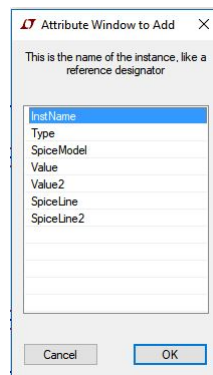
Choose *InstName* and click *OK*, and place the placeholder appropriately
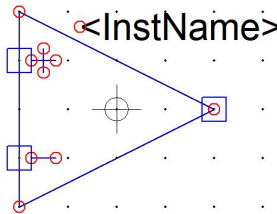
- Save this symbol in the same folder its schematic (*single_stage_opamp.asc*) is saved. Save this symbol as *single_stage_opamp.asy*

- Now create a new schematic. Use the *Add Component* dialog box to navigate to the place where *single_stage_opamp.asc* and *single_stage_opamp.asy* are saved.
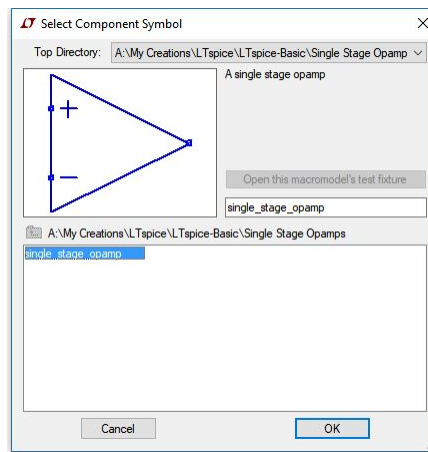
We can see its symbol and description.

- Add the opamp and draw the following schematic. Note that the instance name is *X1*. Also note that the name *Vout* in this schematic has nothing to do with the *Vout* label used in the opamp model schematic.
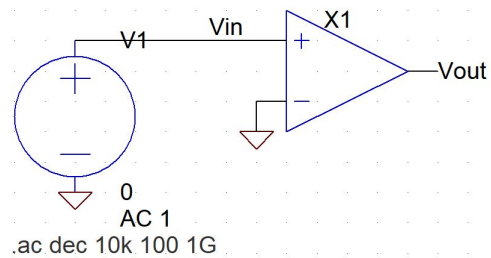

Run an AC simulation to get the bode plot.

- We get the characteristic graph. Note that at the cutoff frequency $(-3dB)$, the phase is $-45°$ as expected.
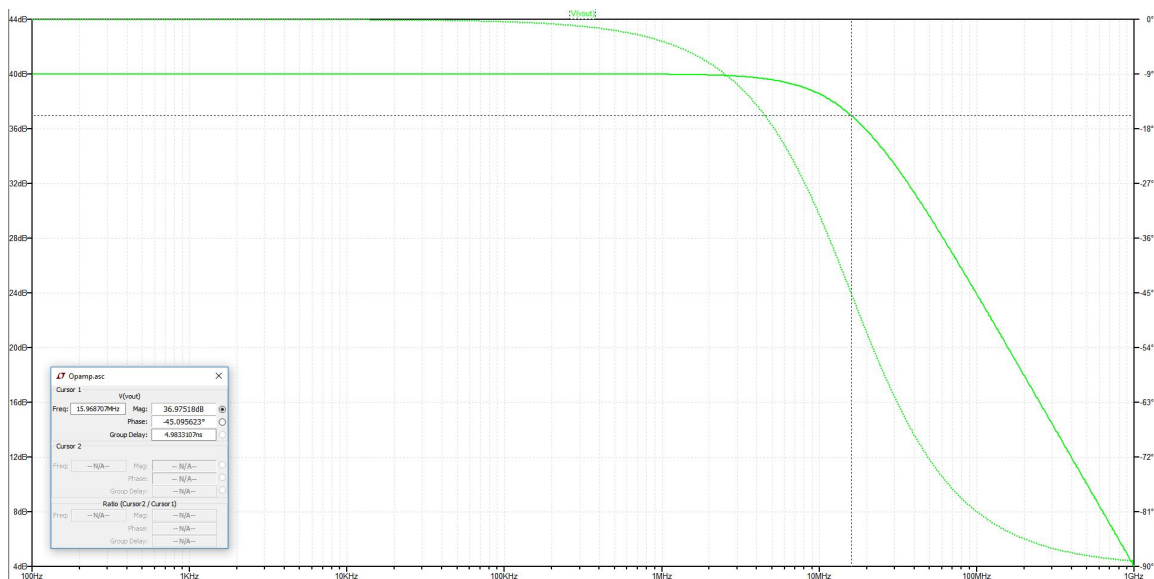
- Now, however if we want to change the value of the components in the model, we have to again modify the *single_ stage_ opamp.asc*. Further, we cannot have two opamps with different specifications. We need a way to parameterize the model.

- To do this, open *single_ stage_ opamp.asc*. Setting a value to *{X}* implies that it is a variable parameter named *X*. Modify the circuit to the one below.
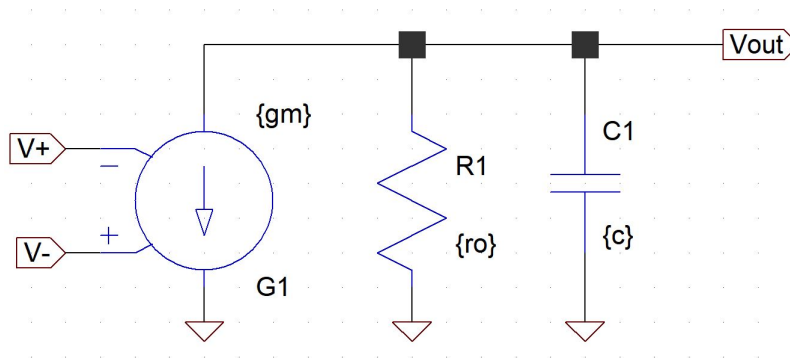
- Like the instance name, we would also like to see the parameters in the schematic. Open *single_stage_opamp.asy*, right-click anywhere in the screen and choose *Attributes->Attribute Window* and add *SpiceLine* to the symbol
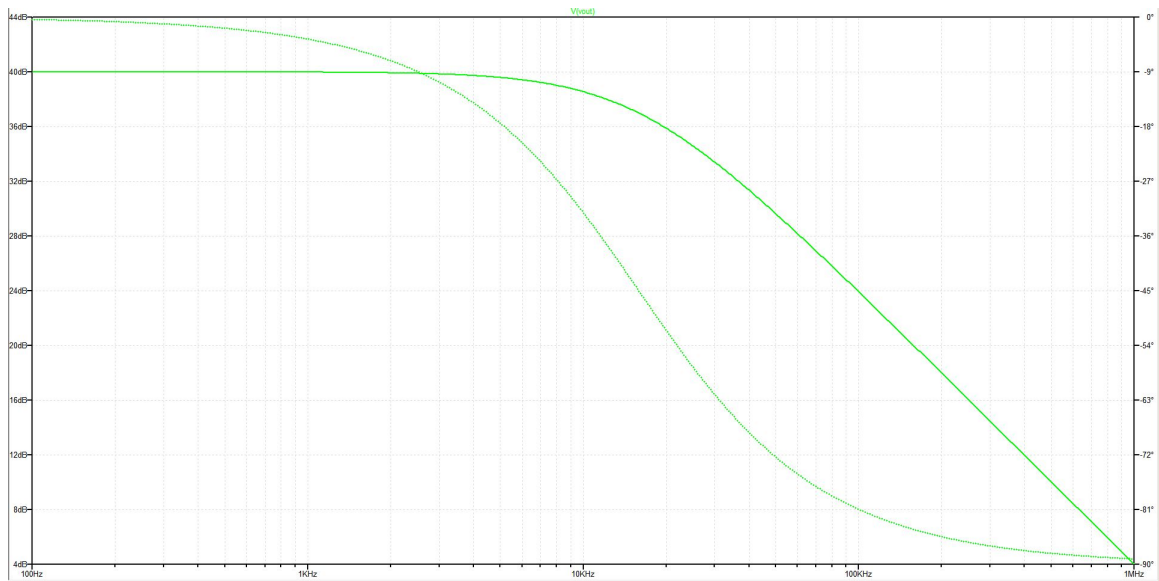
.
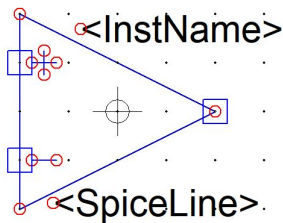
Save both the opamp model symbol and the schematic.

- Now, in the main schematic, right-click the opamp and add the spice line as follows to set the parameters.
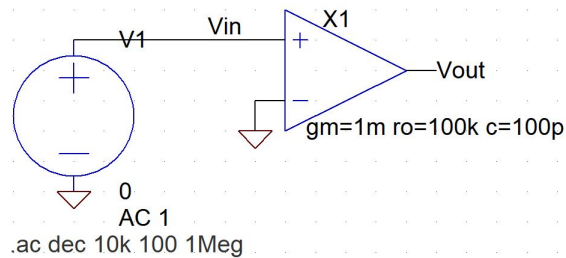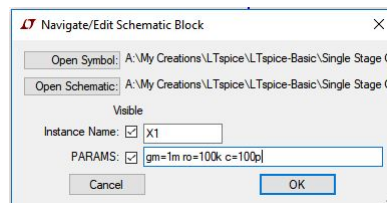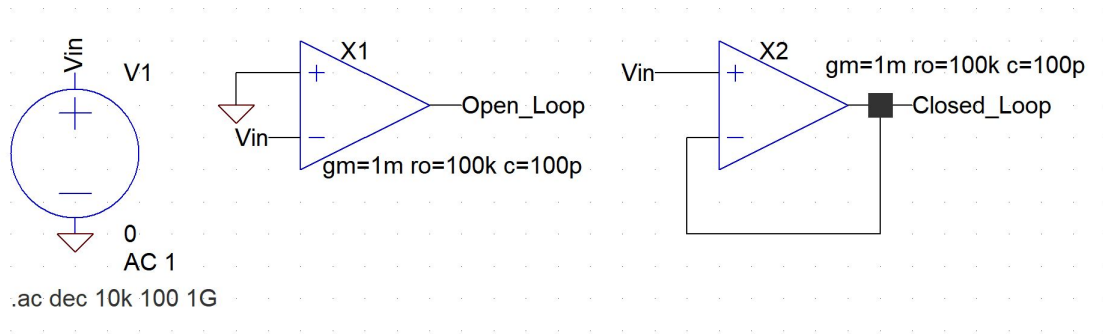

Note that the instance name and spiceline can be moved using the move tool.

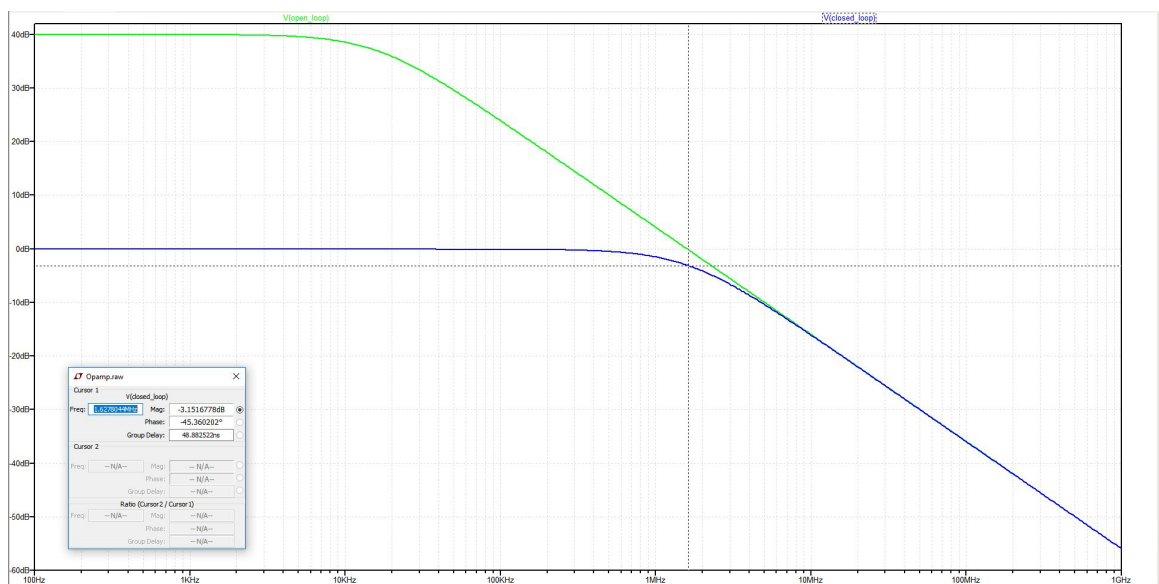- Run an AC Analysis and see the bode plot

The model works as expected.

- Modify the circuit to the one below. Here we analyse our opamp in a unity gain configuration.


- Run an AC analysis and see its bode plot. You can right-click the right axis and choose to not plot the phase. Probe the nodes *Open_ Loop* and *Closed_ Loop*.


Here, you can see that the closed loop bandwidth is greater than the open loop bandwidth. Further, you can see that the closed loop bandwidth is *almost* equal to the unity gain frequency of the open loop transfer function.

- Probe only the *open_loop* node
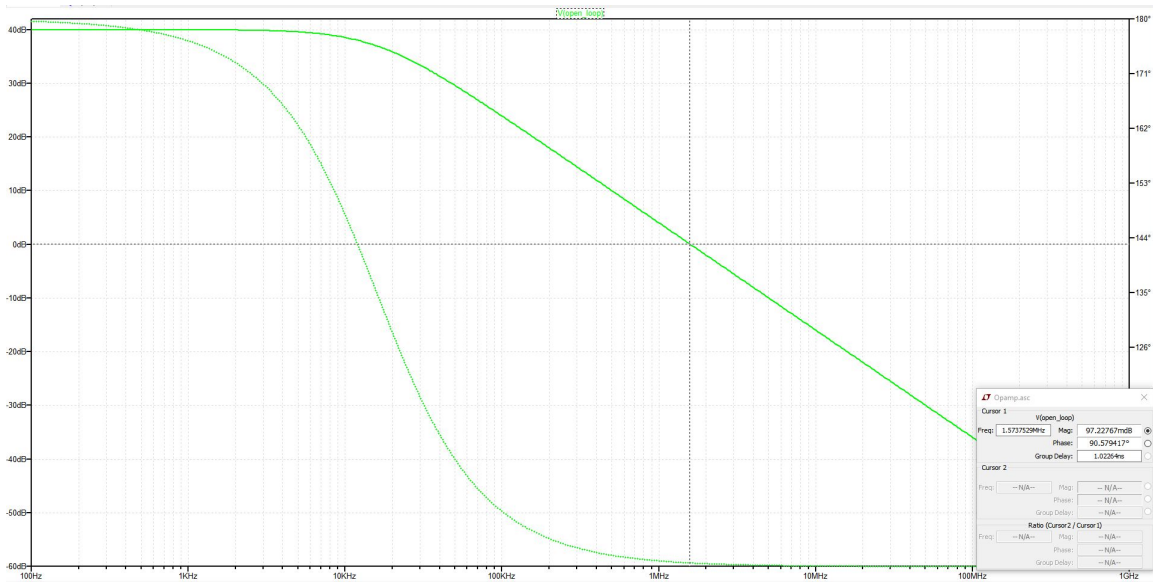
Here, note that LTspice has plotted the phase margin and not the phase.

- Let us analyse the case when this opamp is in a non-inverting gain configuration. We would like to analyse for multiple values of gain in one schematic. To do this, we make use of parameters. Modify the circuit as follows.
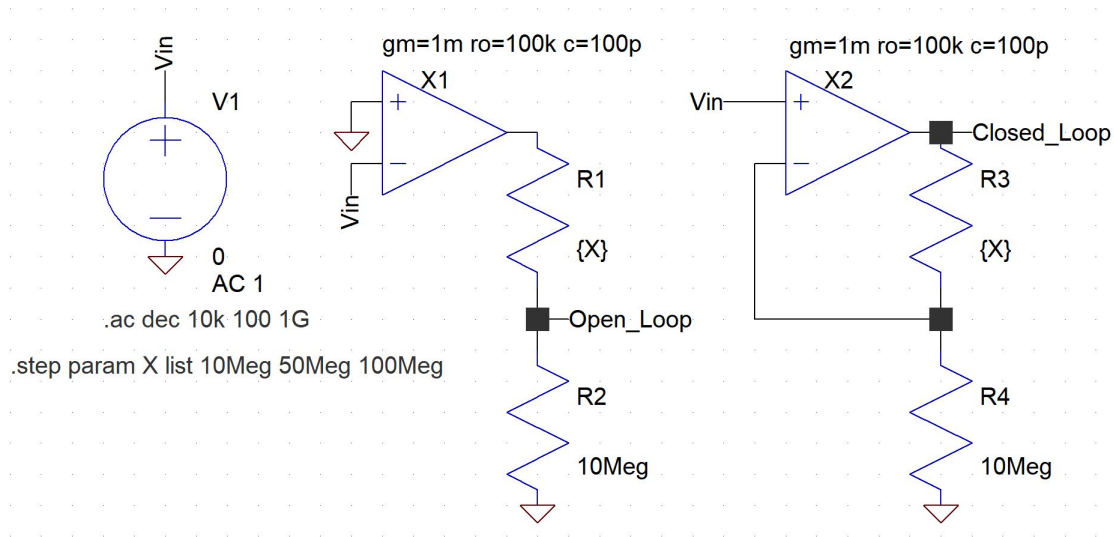
Note that we have chosen large values of feedback resistors to prevent significant loading in the feedback path. The top resistor has a value *{X}*. We use the spice directive *.step* to direct *LTspice* to run the simulation for different values of the parameter given. Add the directive *.step param X list 10Meg 50Meg 100Meg*. This will run the simulation for 3 values of the top resistor.

Also note the place where the loop is broken to get the open loop transfer function.

- Run the AC analysis and probe the node *Open_Loop*
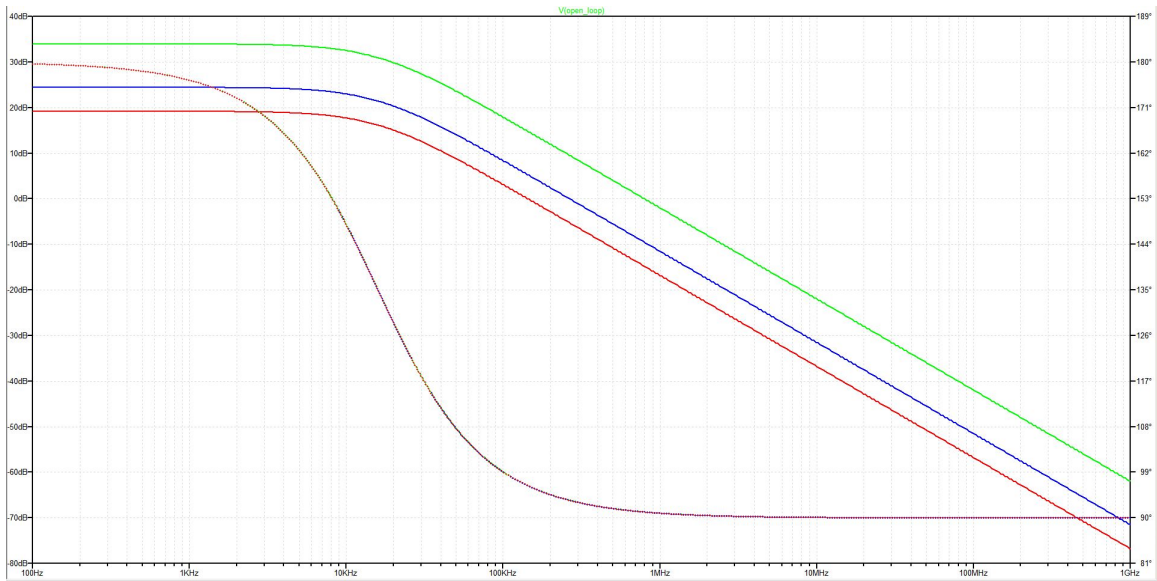
Click on *V(Open_Loop)* to bring the cursor. Use the up and down arrow keys to cycle between the graphs of different gain. To see which resistor value was used for the selected graph, right-click the vertical / horizontal dotted lines of the cursor and a message will pop up. As you can see, for higher closed loop DC gain, the phase margin is much better. Hence, if the opamp used in such a configuration is stable for unity gain, it is definitely stable for larger gains.

- Probe the node *Closed_Loop*
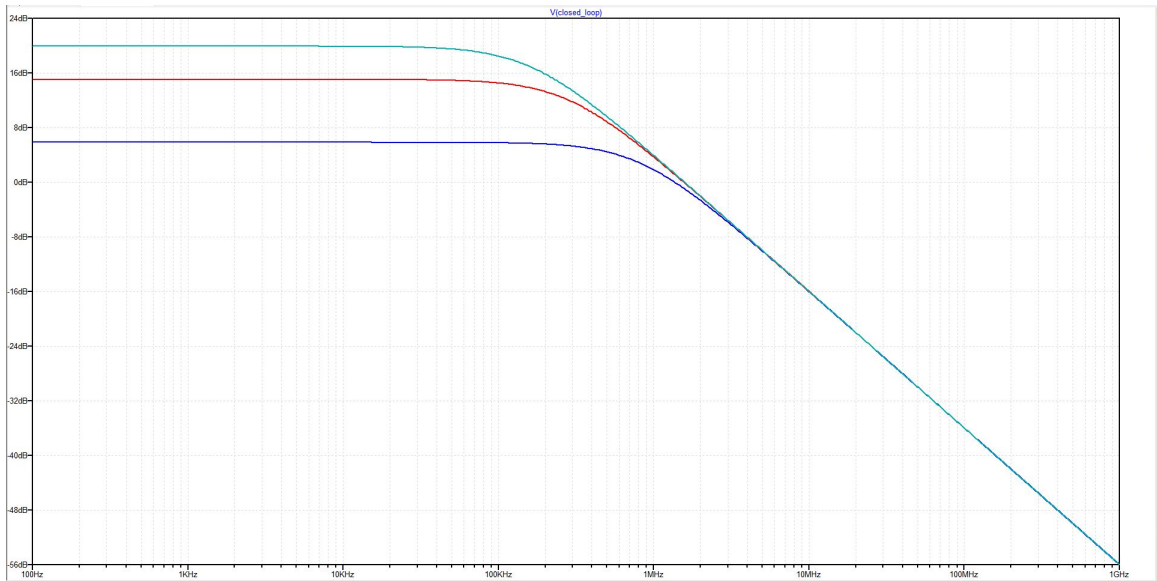
Note that for higher closed loop DC gain, the bandwidth falls. Also note that all the lines join together at high frequencies. This can be used to show that the gain bandwidth product is constant for this opamp in this configuration.