# **Bug #3: Log Scale (Fixed)**

**Bug/Issue:** [Minor ticks on log-scale colorbar are not cleared #8358](https://github.com/matplotlib/matplotlib/issues/8358)

**Estimated Hours:**

*- Explore and create a solution (22 h)*

*- Implement solution (11 h)*

*- Testing/validation (12 h)*

*- Code Review (4 h)*

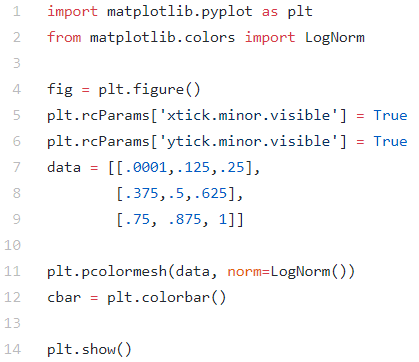
*- Documentation (1 h)*

*- Total: 50 h*

**Description:**

When creating a colorbar with a logarithmic scale, its minor ticks are already shown by default. However, if the user’s rcParams values ‘xtick.minor.visible’ or ‘ytick.minor.visible’ is true so that minor ticks are always visible by default, then linear scale minor ticks will unexpectedly be present in addition to the log scale ticks.

Example:





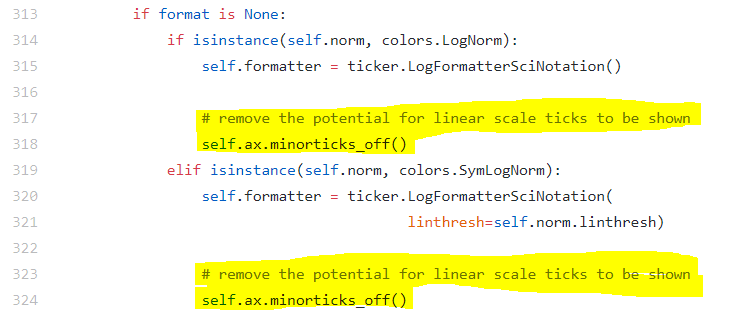
**Solution:**

[Updated Forked Repo](https://github.com/gravitybear/matplotlib)

The affected file is ***colorbar.py*** and its updated version with our solution can be found in ***solutions/logscale/colorbar.py***.

For our solution, we spent a lot of time thinking of an effective way of combating the issue. We started by looking into fixing the original implementation of LogLocator. In the responses to the issue, a contributor mentioned that the ticks produced by LogLocator are actually emulated using major ticks, which is the cause of its odd behaviour. We wanted to edit the class such that log scale minor ticks are treated like the others (i.e. hidden by default, affected by Axes.minorticks\_on() or Axes.minorticks\_off()). However, we decided that this solution could break existing uses and dependencies, as its implementation has been around for years.

Instead, we looked into removing the minor ticks while the Colorbar is being created. Although our solution was simple, we came to it through hours of trial and error. Our goal was to remove the unwanted ticks, so we started by appending different methods of the Axis, Axes and Colorbar classes that were related to minor ticks to the end of our test script to see which methods did the job. While browsing the axes/\_base.py, we noticed that it had a method called minorticks\_off() which says it removes minor ticks from the axes. So, we tested this method, and it produced the expected result. Our next goal was to find a way to identify which types of ticks were being used. We tried checking if the locator was LogLocator, or if the scale was ‘log’, but due to the odd implementation of log scale ticks, neither of these worked. Then, we noticed that there was an if-statement in the \_\_init\_\_ method of ColorbarBase that checked if it was given a log norm. We plugged minorticks\_off() into there, and it produced the desired result. After successfully testing it over a series of existing and new test cases to make sure that it does not break existing functionality, we decided that it is a good solution, which is shown below.



**Testing:**

Once again, image comparison was used to test the solution. The new tests are located in ***solutions/log\_scale/tests/***. There, the file ***test\_logscale.py*** contains the new test cases written for our solution. Since the bug and its solution is rooted in the Colorbar, these tests should be added to matplotlib's existing ***test\_colorbar.py***. This updated ***test\_colorbar.py*** can be found alongside ***test\_logscale.py***, in ***D4/solutions/log\_scale/tests/***, and should replace matplotlib's ***test\_colorbar.py*** in the event of a merge.

Since the issue bug appears with a log scale colorbar where the ‘xtick.minor.visible’ or ‘ytick.minor.visible’ rcParam values are set to true, we made a number of graphs reproducing this situation with different parameters. The test cases used different ranges of data sets and also made sure to verify that the bug is fixed for horizontal colorbars as well as the default vertical ones. Additionally, both the LogNorm() and SymLogNorm() parameters for the pcolormesh graph are tested, since they both implement logarithmic scales.

The resulting images of our tests, shown below, can be found in ***solutions/log\_scale/tests/test\_colorbar/*** and should be added to matplotlib's ***lib/matplotlib/tests/baseline\_images/test\_colorbar/*** during a merge. As one can see, there are no longer any undesired linear ticks on the colorbar despite the visibilities of the minor ticks being set to true (this is evident because all of the colorbar ticks are uniform in length – that is, there are no minor ticks). The updated ***test\_colorbar.py*** was also run on our fix and all the tests have passed.

|  |  |
| --- | --- |
| *simple\_log\_cbar.png* | *simple\_log\_cbar\_horizontal.png* |
| *dense\_log\_cbar.png* | *dense\_log\_cbar\_symlognorm.png* |

**Confidence in Solution:**

We believe our solution works well because it solves the bug that the user was faced with. The purpose of rcParams is to add customizability to matplotlib so that users can easily set default styles for all their plots. In colorbars with many minor ticks, it is difficult to notice ticks that should not be there. The user that encountered this bug, and possibly many others have their ‘xtick.minor.visible’ and ‘ytick.minor.visible’ values set to true, and by fixing this bug, they will not see the unexpected linear scale ticks, and produce plots with log scale minor ticks correctly displayed on their colorbars.