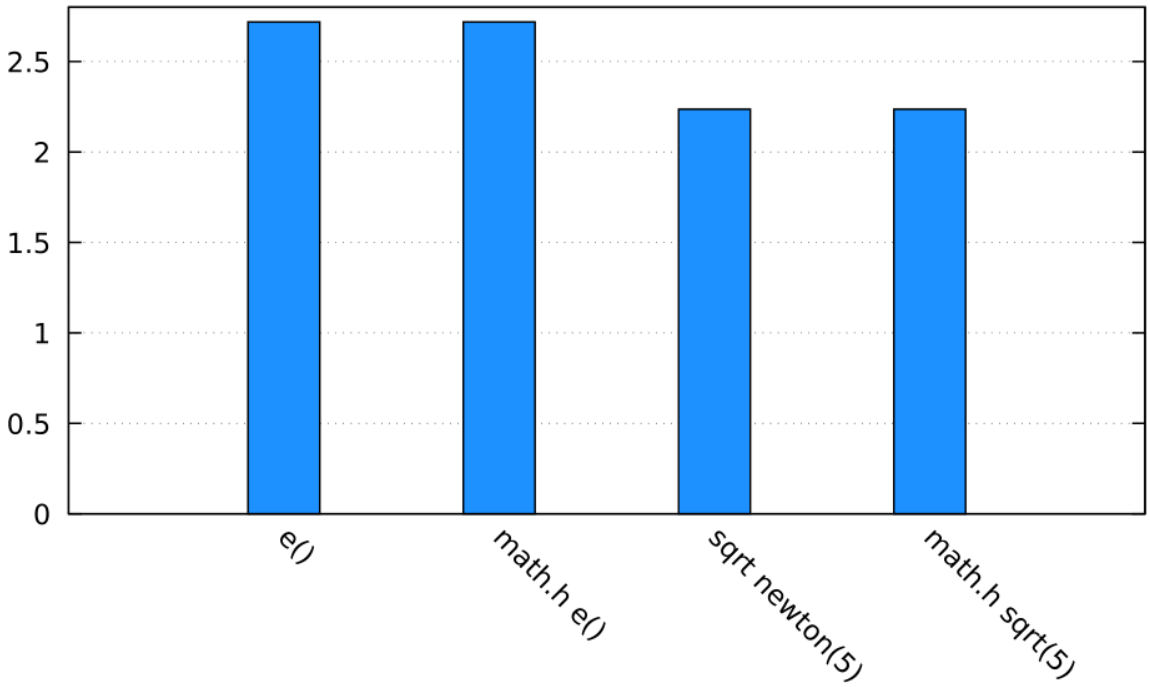
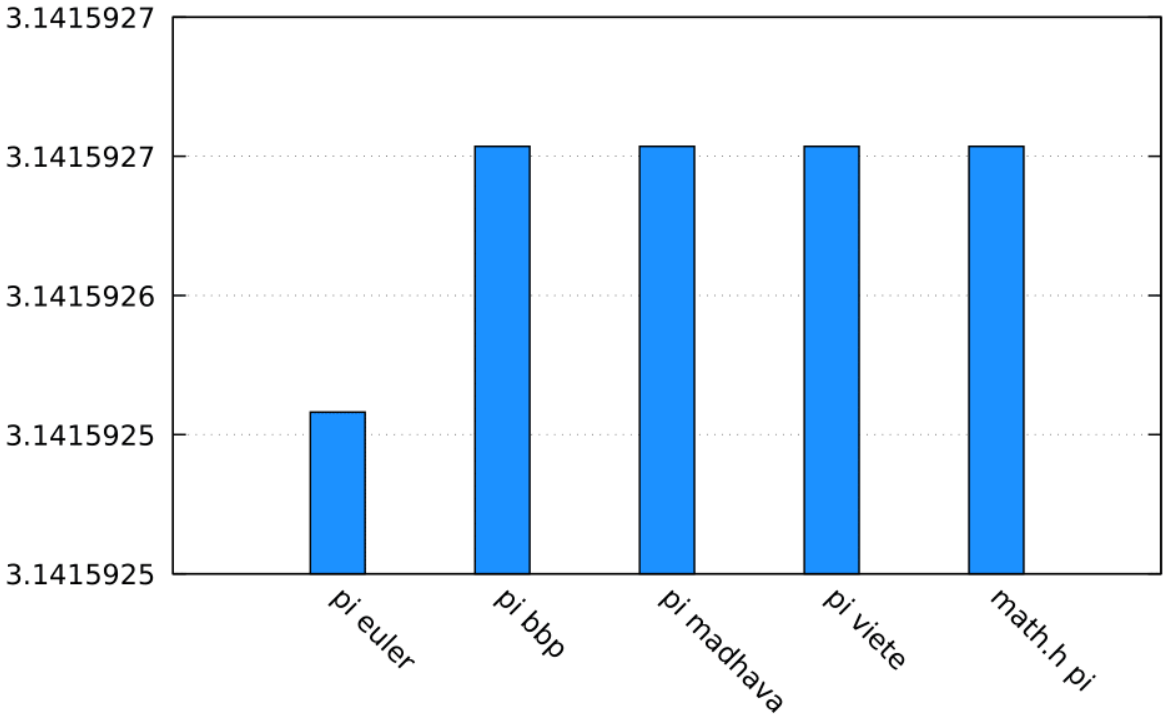
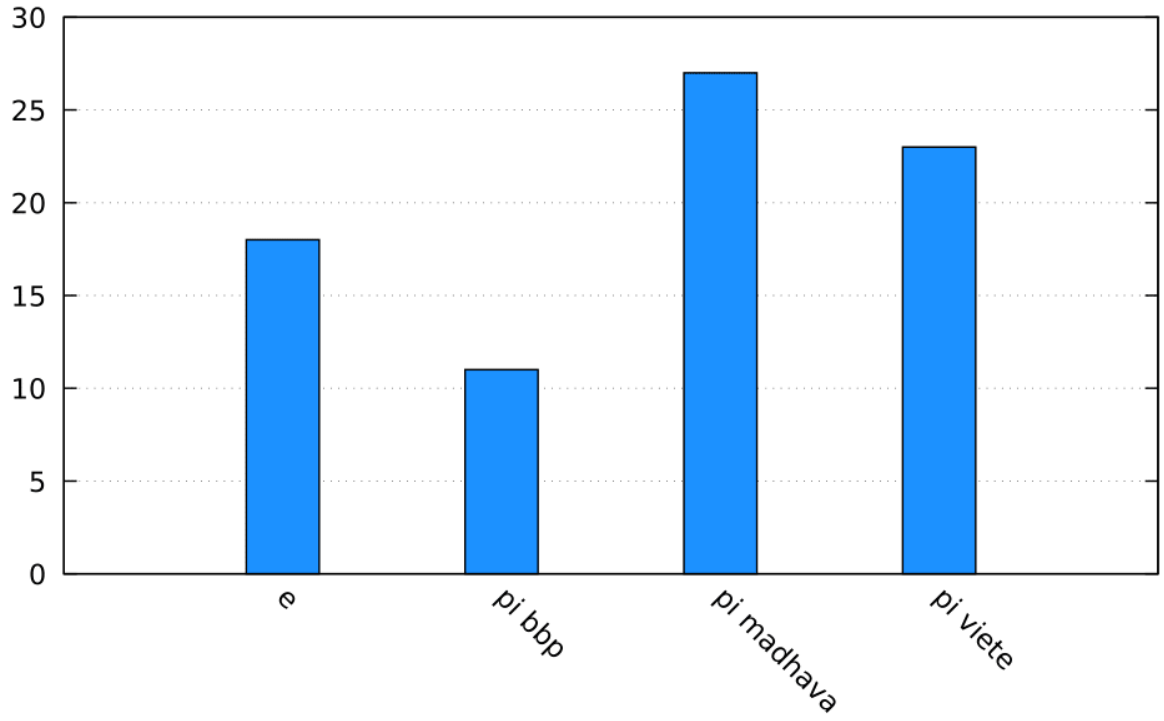


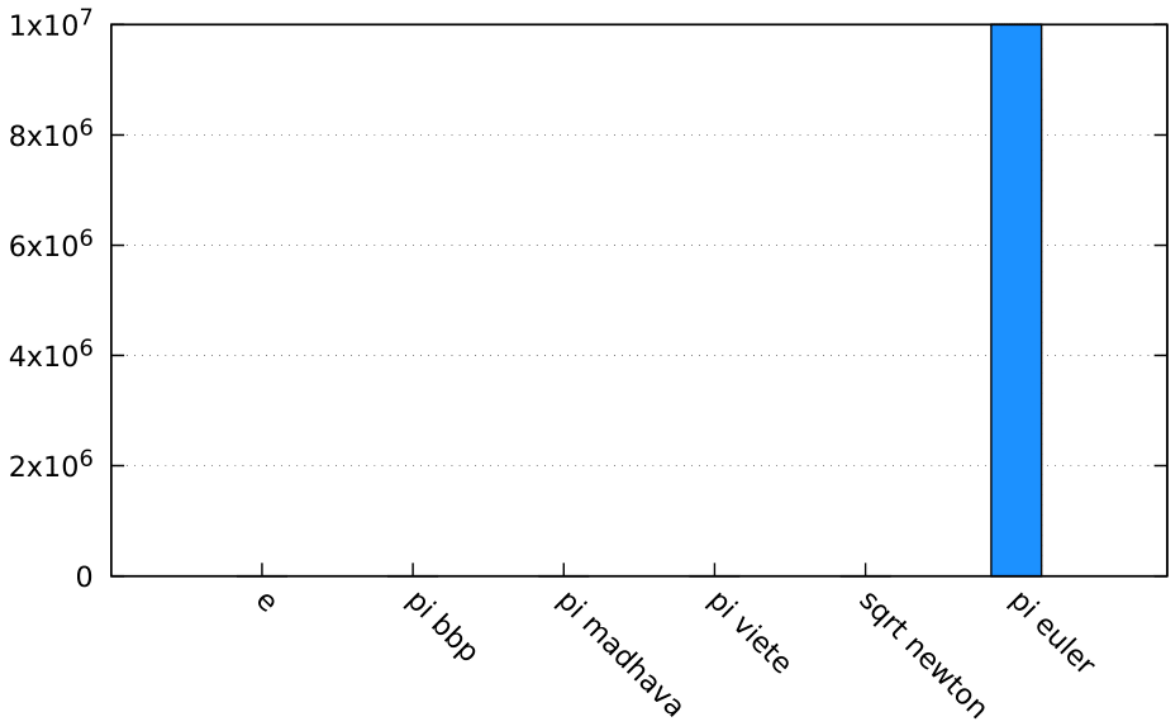
To the decimal I estimated with the %16.15lf print I was told to use, my e() and math.h's M_E were equal, as well as my srt_newton() and math.h's sqrt(). This was consistent with the values from the given resource binary. My sqrt(0) was different from the binary, since I decided to hard code a sqrt(0) = 0 instead of failing to approximate it.



The same was true of my pi functions. bbp was the only one who's error showed up as 0, but the others' error looks invisible compared to euler. All the error values were consistent with the binary.



The difference in the amount of terms was interesting. alongside the error, it shows how bbp was overall the best method we coded. These were all the same as the resource binary. I didn't include sqrt_newton here, but its terms varied in the same range as the binary, 1 to 7.



When you include euler on this list, none of the other terms even appear, since euler adds 10 million terms.