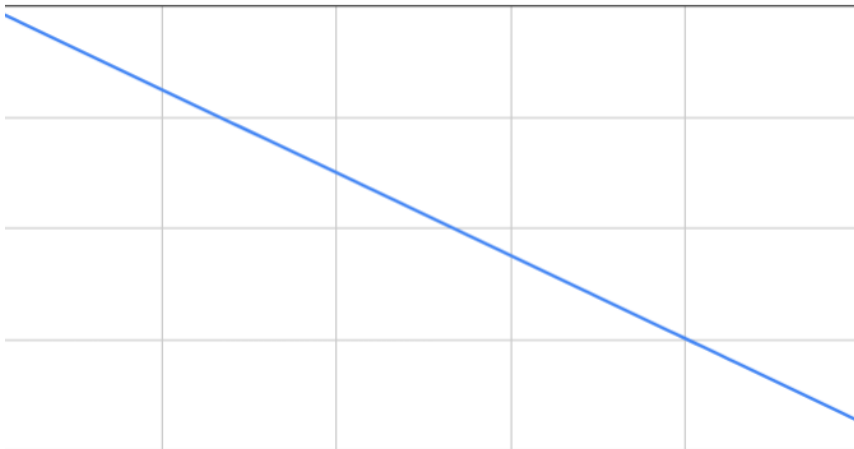


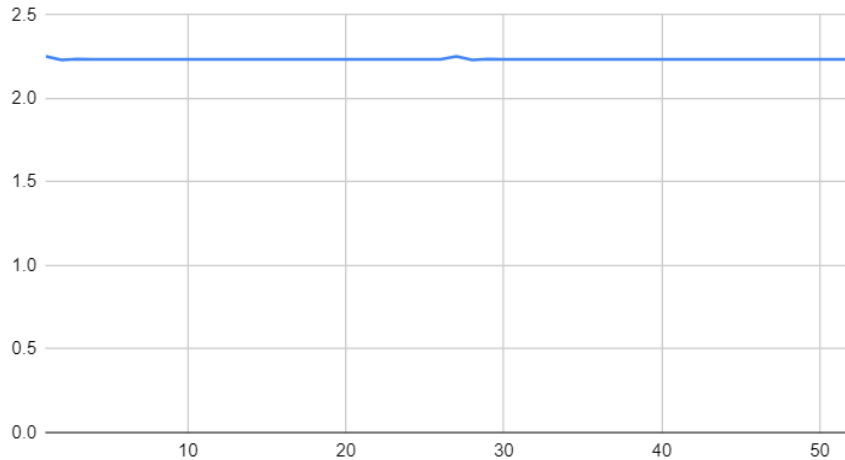
This graph compares how close each term of e is to the `math.h` `M_E`. We can see that with the 1st term calculated, there is a difference of 0.7, meaning that my first calculated term is 0.7 away from the `math.h` `M_E`. We can see that with more terms calculated, we match the `math.h` `M_E`.

Euler's solution vs `M_PI`



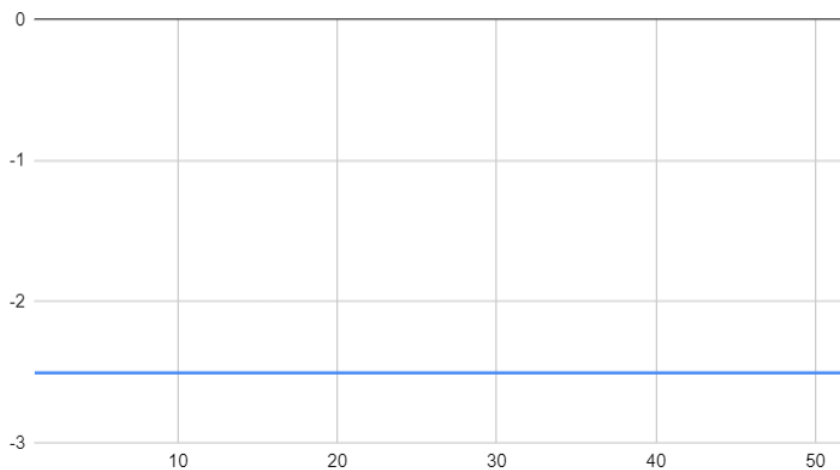
This graph shows the difference for each term calculated vs `M_PI`. As you can see it's linear, each term slowly gets more accurate and approaches `M_PI`. Except in my code, it stopped calculating when there was a difference of 0.000000095493891. I think that's because of how accurate we can be using Euler's solution.

Each Madhava term vs M_PI



In this graph we see the difference of each term calculated using the madhava formula. The difference seems constant, I think it is constant because we have not multiplied by square root 12 to reach M_PI.

Each Viète term vs M_PI



In this graph we see the difference of each term with M_PI, again it is constant because we are multiplying each term with something smaller. For whatever reason my implementation need 54 terms while the one in resources only needs 24, yet they give the same results.