4.1 Identify gaps in the Testing Process

The main gaps in the testing come from the purely functional nature and the randomly generated test points.

The purely functional nature of my testing means that there aren't any formal code coverage measures. This means it not possible to state how much of the implementation was exercised and tested. Adding formal code coverage metrics would have allowed for more targeted tests, as it would be possible to see how much of the implementation was already covered and what had not been covered.

The randomly generated test points mean that edge cased were potentially not covered. Repeating the tests 10 times with new randomly generated points each ensures wider coverage than just testing with one set of points, but there is still no guarantee that edge-cases are covered and that the points generated reflect real world usage. Hand selecting a few sets of points to cover edge and extreme cases on top of the randomly generated sets would have led to much better coverage of scenarios.

A gap is also found in the performance testing. While the test used does meet the 30-second requirement, no insight is provided into any potential bottlenecks or how performance is spread. This prevents any further analysis of performance, and provides no knowledge or insight on how performance could be improved,