

**Answer.** It is obvious that the absolute tolerance of 0.001 is extremely rough when it comes to compute the square root of a relatively small number. Let's take 0.00001 for example:

Instead of putting out 0.0031622776601683794, which is legitimate. The original compound procedure `sqrt` gives a wrong answer with a tolerance over 900%.

The odd thing happens here is because when the radicand becomes relatively giant, the interpreter might fail to represent the difference between `guess` and `x` within the tolerance of 0.001 with its limited precision. But the interpreter won't jump out the compound procedure `sqrt-iter` until the predicate `good-enough?` is sufficed. Thus this is contradictory and the interpreter drops into an endless sequence of recursive calls.

Having the new **good-enough?** conditional defined above, we can easily rewrite the original square-root procedure as the following:

Now let's do some test on the new `sqrt` procedure as before:

By comparing the performance of the new `sqrt` procedure and the original version from the book. We are now sure that the new version of `sqrt` procedure works better for both small and large number.