

[Home](#)[Lesson-4.1](#)

Lesson 3.6

Lesson 3.6

[Math and Programming](#)[Limits](#)[Recurrence relation](#)[Rational Approximation](#)

Math and Programming

Before closing this chapter, let us spend some time at the intersection of mathematics and programming.

Limits

Consider the following number:

$$\sqrt{2} - 1$$

It is known that $1 < \sqrt{2} < 2$. From this, it follows that $0 < \sqrt{2} - 1 < 1$. Now, consider the following sequence:

$$a_n = (\sqrt{2} - 1)^n$$

As n becomes very large, the values in this sequence will become smaller and smaller. This is because, if you keep multiplying a fraction with itself, it becomes smaller and smaller. In mathematical terms, the limit of this sequence as n tends to infinity is zero. Let us verify this programmatically:

```
1 import math
2 n = int(input())           # sequence length
3 CONST = math.pow(2, 0.5) - 1 # basic term in the sequence
4 a_n = 1                   # zeroth term
5 for i in range(n):
6     a_n = a_n * CONST      # computing the nth term
7 print(a_n)
```

Try this out for a few values of n . For $n = 100$, the value is 5.27×10^{-39} , which is so small that for all practical purposes, it is as good as zero.

Recurrence relation

Now, here is another fact. For every number n , there are unique integers x and y such that:

$$(\sqrt{2} - 1)^n = x + y \cdot \sqrt{2}$$

For $n = 1$, this is obvious: $x = -1, y = 1$. What about higher values of n ? We can prove this using mathematical induction. The following is a sketch of the inductive proof. If

$(\sqrt{2} - 1)^n = x_n + y_n \cdot \sqrt{2}$, then:

$$\begin{aligned} (\sqrt{2} - 1)^{n+1} &= (x_n + y_n \cdot \sqrt{2}) \cdot (\sqrt{2} - 1) \\ &= (2y_n - x_n) + (x_n - y_n) \cdot \sqrt{2} \\ &= x_{n+1} + y_{n+1} \cdot \sqrt{2} \end{aligned}$$

The equation given above defines what is called a recurrence relation: each new term in the sequence is a function of the preceding terms. In this sequence we have $x_1 = -1, y_1 = 1$. For $n > 0$, the pair of equations given below forms the recurrence relation:

$$\begin{aligned} x_{n+1} &= 2y_n - x_n \\ y_{n+1} &= x_n - y_n \end{aligned}$$

Loops are useful tools when it comes to computing terms in such sequences:

```
1 n = int(input())    # sequence length
2 x_n, y_n = -1, 1    # x_1 and y_1
3 for i in range(n - 1):
4     x_n, y_n = 2 * y_n - x_n, x_n - y_n
```

Rational Approximation

This in turn provides a way to approximate $\sqrt{2}$ using rational numbers:

$$\sqrt{2} \approx \frac{-x_n}{y_n}$$

As n becomes large, this approximation will become increasingly accurate. For example, here is an approximation after 100 iterations. It is accurate up to several decimal places!

$$\frac{228725309250740208744750893347264645481}{161733217200188571081311986634082331709}$$

Is any of this useful? I don't know. But honestly, who cares? We don't do things because they are useful. We do them because they are interesting. And all interesting things will find their use at some point of time in the future.