1 Floating Point Exercises

Exercise 1. *Approximating Pi*

Pi (π) is an irrational number describing the relationship between a circles diameter and its circumference. Because it is irrational, it has infinite decimal places when represented as a floating point number. There are many different methods to approximate this floating point number up to a certain accuracy. In this exercise we will make use of the Gregory-Leibniz series:

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{2n+1}$$

We can bring π into this sum, by replacing arctan with an expression containing π. We know that $\tan(\frac{\pi}{4}) = 1$ and therefore $\arctan(1) = \frac{\pi}{4}$. Plugging this back into the series yields:

$$\begin{aligned} arctan(1) &= \sum_{n=0}^{\infty} \frac{(-1)^n 1^{2n+1}}{2n+1} \\ &= \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \\ &= 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots = \frac{\pi}{4} \end{aligned}$$

Rewriting and multiplying by 4 leaves us set up for the programming part.

$$4 \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \pi$$

To approximate π, we could now calculate more and more elements of this sum, depending on how accurate we want our approximation to be.

a) Create a program that approximates the value of π up to the 2'nd decimal point, so 3.14.

b) Modify the program so that the user can ask for a custom accuracy. Example:

```
What precision do you want for PI?  0.0001
Calcualted PI: 3.1414926535900345
Deviation from actual PI: 9.99999997586265E-5
Iterations it took: 10000.0
```

Be careful with the input for the accuracy here, as the run time get very big very quickly. The accuracy of 0.00001 took around 3 minutes to compute on my machine.