GumbelTalks LaTeX Intro

An example exercise sheet using LaTeX

Leon Rauschning

15/4/23

1 Solve a linear equation

In exercise 1, we will demonstrate how to solve a linear equation.
The given equation is of form

$$y = mx + t \tag{1}$$

, with m being the *slope* and t the *intercept*. We solve for x:

$$\begin{aligned} y &= mx + t \\ y - t &= mx \\ x &= \frac{y-t}{m} \end{aligned}$$

With $y = 6$, $t = 2$ and $m = 2$ given in the exercise, we can show that

$$x = 2$$

holds. The same result could also be obtained using `scipy.linalg.solve` **or any other equation solving software**. □

See also Fig. 1 for a graphical illustration of the solution of Eqn. 1.

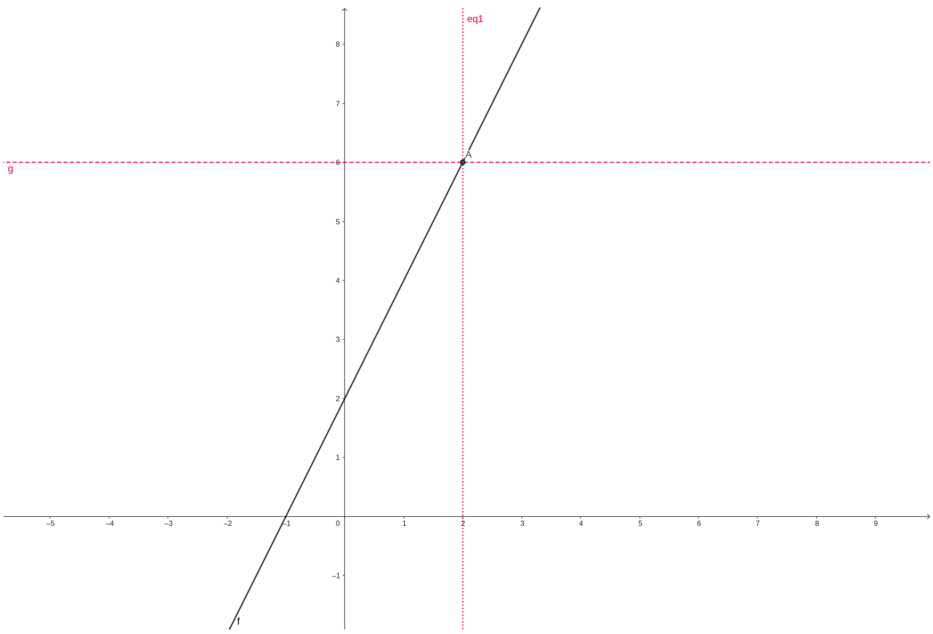

Figure 1: Graphical solution to Eqn. 1 using GeoGebra

2 Everyone loves pie!

In exercise 2, we will demonstrate an approximation for π. We introduce the Leibniz series $\mathcal{L}$:

$$\mathcal{L} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

This series can be shown to converge to $\frac{\pi}{4}$ by forming it into a geometric series:

$$\begin{aligned} \mathcal{L} = & \quad \textstyle\sum_{n=1}^{\infty} \frac{(-1)^n}{2n+1} \\ = & \quad \textstyle\sum_{n=0}^{\infty} (-1)^n \int_0^1 x^{2k} \, \mathrm{d}x \\ = & \quad \textstyle\int_0^1 \sum_{n=0}^{\infty} (-x^2)^n \, \mathrm{d}x \end{aligned}$$

Applying the formula for geometric series to the term inside the integral, we are able to eliminate the infinite series from this expression:

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

Ultimately leaving us with a simple integral:

$$\mathcal{L} = \int_0^1 \frac{1}{x^2+1} \, \mathrm{d}x = [\arctan(x)]_0^1 = \arctan(1) = \frac{\pi}{4}$$

Thus, we have shown that the Leibniz series can be used as a simple way to compute an approximation of π. An example for the approximation of π in this manner is given in Table 1. □

Table 1: This table shows the approximation of π by the Leibniz sum as more terms are added to the sum. Correct digits are written in bold. The values were calculated using python: `print(4*sum(map(lambda x: ((-1)**x)/((2*x)+1), range(n))))`

Number of terms (n)	Value
1	4
10	**3**.0418396
100	**3.1**315925
1000	**3.14**05926
10,000	**3.141**4929
100,000	**3.1415**826
1,000,000	**3.14159**16