Fun with $\mathrm{sinc}(t)$[1]

1 Introduction

The *unnormalized sinc*[2] function, also called *sampling function*, is defined as

$$\mathrm{sinc}(x) = \frac{\sin(x)}{x}.$$

Using L'HÔPITAL's rule, the value of $\mathrm{sinc}(0)$ can be determined easily since the numerator and denominator have the limit 0 and the first derivative of both also exists:

$$\lim_{x\to 0}\mathrm{sinc}(x) = \lim_{x\to 0}\frac{\sin(x)}{x} = \lim_{x\to 0}\frac{\cos(x)}{1} = 1.$$

Figure 1 shows the graph of $\mathrm{sinc}(x)$ and its normalised variant.[3]

This function occurs in many contexts – its normalized variant is the FOURIER transform of the *rectangle* function[4]

$$\mathrm{rect}\left(\frac{t}{a}\right) = \begin{cases} 0 & \text{if } |t| > \frac{a}{2} \\ \frac{1}{2} & \text{if } |t| = \frac{a}{2} \\ 1 & \text{if } |t| < \frac{a}{2} \end{cases}$$

, thus it also describes the amplitudes of light diffracted at a slit, it even has connections to prime numbers RIEMANN's ζ-function, it can be used to reconstruct signals from sampling data, etc.

This application note shows two approaches for generating $\mathrm{sinc}(x)$ for $x > 0$ (and not too large). In both cases x is replaced by the machine time t, which is generated by integrating over a (small) constant.

[1] The author would like to thank Dr. CHRIS GILES for fruitful discussions and his meticulous proofreading.

[2] The *normalized sinc* function is defined as $\mathrm{sinc} x = \frac{\sin(\pi x)}{\pi x}$.

[3] Source: By GEORG-JOHANN - Own work, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=17007237`.

[4] Also called the HEAVISIDE *Pi* function.

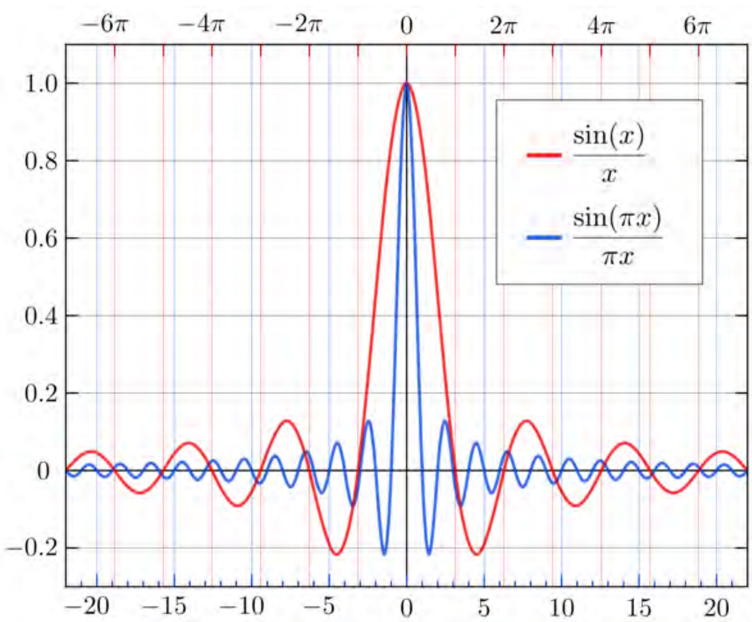

Figure 1: Graph of $\mathrm{sinc}(x)$

2 Direct approach

The first approach is straightforward: Generate $\sin(t)$ and divide it by t. Mathematically this works fine, although the division may misbehave for very small values in an analog computer.

Generating a sine function is basically the "hello world" of analog computing and typically done by solving $\ddot{y} = y$. Since the numerator t is limited to the interval $[0, 1]$, there is no need to employ any form of amplitude stabilisation.

The straightforward implementation is shown in figure 2, while figure 3 shows the corresponding result. The division is implemented using an *open amplifier* with a multiplier in its feedback path.[5]

[5] Cf. [Ulmann 2023, p. 76]. It may be necessary to add a small capacitor (around 100 pF maximum) between the output of the amplifier and its summing junction to stabilise this subcircuit.

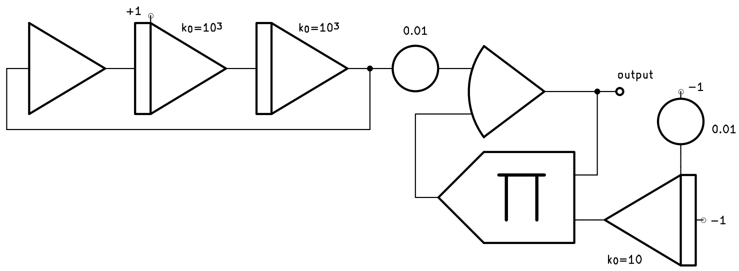

Figure 2: Analog computer setup for equation (2)

In order to obtain a number of oscillations of $\operatorname{sinc}(t)$ as the output t, should run from some small ε to about 100, which is obviously impossible, given the machine interval of $[-1, 1]$. The "trick" is to restrict t to $[\varepsilon, 1]$ an upscaling it during division. Of course, the denominator cannot be $100t$ as this would exceed the machine unit interval. Instead, the numerator is multiplied by $\frac{1}{100}$. The integrator yielding t should start at a very small positive value instead of 0.

As one can see, the result deviates substantially from $\operatorname{sinc}(t)$ near 0 as a division of the form $\frac{\varepsilon_1}{\varepsilon_2}$ with small ε_i not necessarily yields a result close to 1.

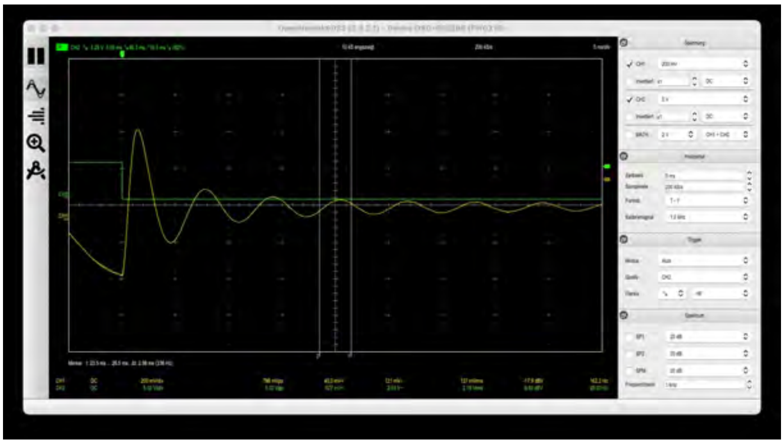

Figure 3: Result of the program shown in figure 4

3 Indirect approach

This second approach is a little bit more involved as it is based on deriving a differential equation with $\operatorname{sinc}(\tau)$ as its solution (given suitable initial conditions). To derive such a DEQ we need the first and second derivatives:

$$\begin{aligned} y &= \frac{\sin(t)}{t} \\ \dot{y} &= \frac{\cos(t)}{t} - \frac{\sin(t)}{t^2} \\ \ddot{y} &= -\frac{\sin(t)}{t} + 2\frac{\sin(t)}{t^3} - 2\frac{\cos(t)}{t^2} \end{aligned} \tag{1}$$

Combining these three equations the following DEQ can be derived

$$t\ddot{y} + 2\dot{y} + ty = 0,$$

yielding

$$\ddot{y} = -\frac{2\dot{y} + ty}{t}. \tag{2}$$

The corresponding two initial conditions can be derived in a straightforward way:

$$y(0) = \lim_{t\to 0} \frac{\sin(t)}{t} = 1$$

as shown above. Visual inspection suggests $\dot{y} = 0$ which can be shown easily using short TAYLOR approximations for $\sin(t)$ and $\cos(t)$:

$$\sin(t) = t - \frac{t^3}{6} + \mathcal{O}(t^4) \tag{3}$$

$$\cos(t) = 1 - \frac{t^2}{2} + \frac{t^4}{24} + \mathcal{O}(t^5) \tag{4}$$

Using (1) in conjunction with (3) and (4) yields

$$\dot{y}(0) = \lim_{t\to 0} \frac{\cos(t)}{t} - \frac{\sin(t)}{t^2} \approx \lim_{t\to 0} \frac{1}{t} - \frac{t}{2} + \frac{t^3}{24} - \frac{1}{t} + \frac{t}{6} = 0.$$

Using (2) with $y(0) = 1$ and $\dot{y} = 0$ can be directly transformed into an analog computer setup as shown in figure 4. The corresponding result is shown in figure 5. t is created and treated exactly as described above.

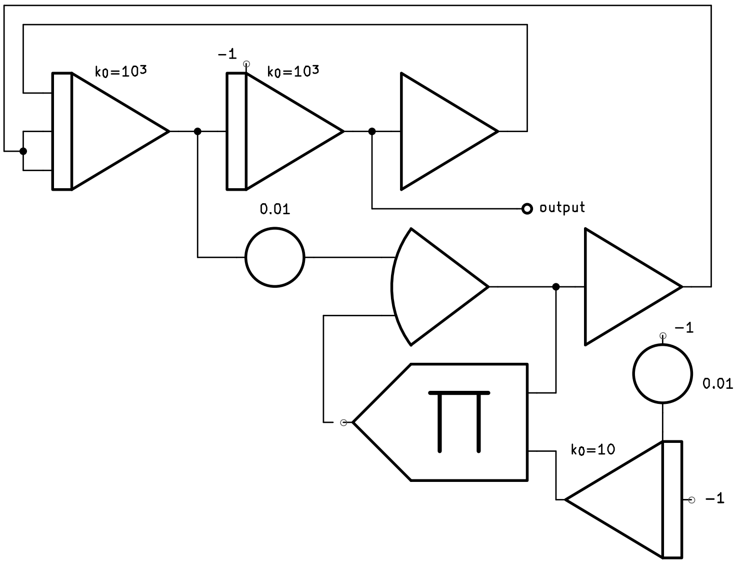

Figure 4: Analog computer setup for equation (2)

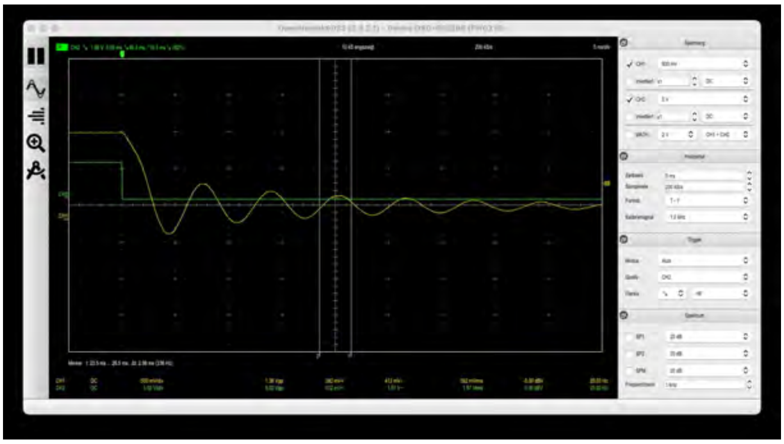

Figure 5: Result of the program shown in figure 4

4 Conclusion

While both solutions do not behave perfectly near 0, the second approach yields a much better result than the straightforward solution. The difference between both $\operatorname{sinc}(t)$ implementations is shown in figure 6. The overall setup, implementing both approaches at once, is shown in figure 7.

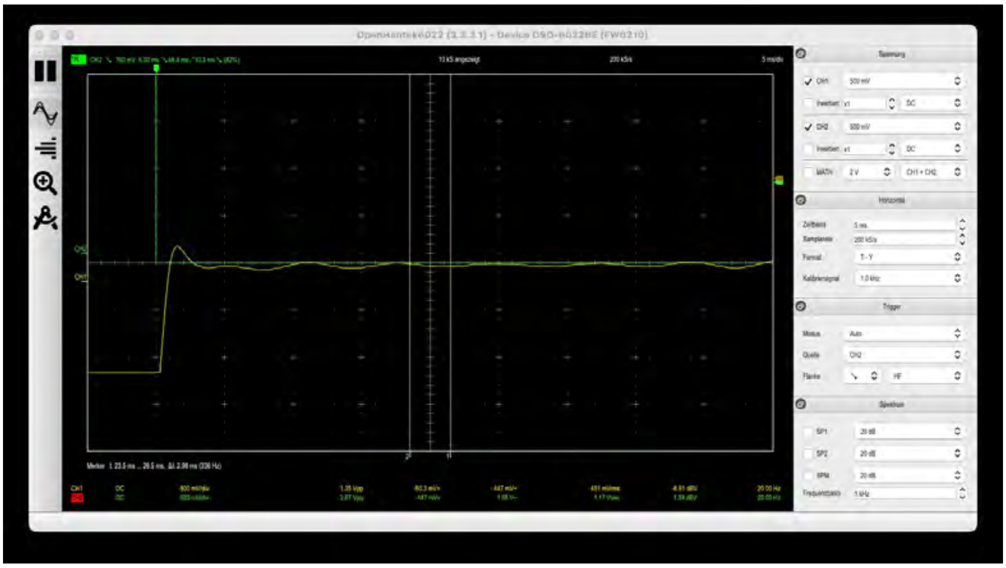

Figure 6: Difference between the results obtained by both methods

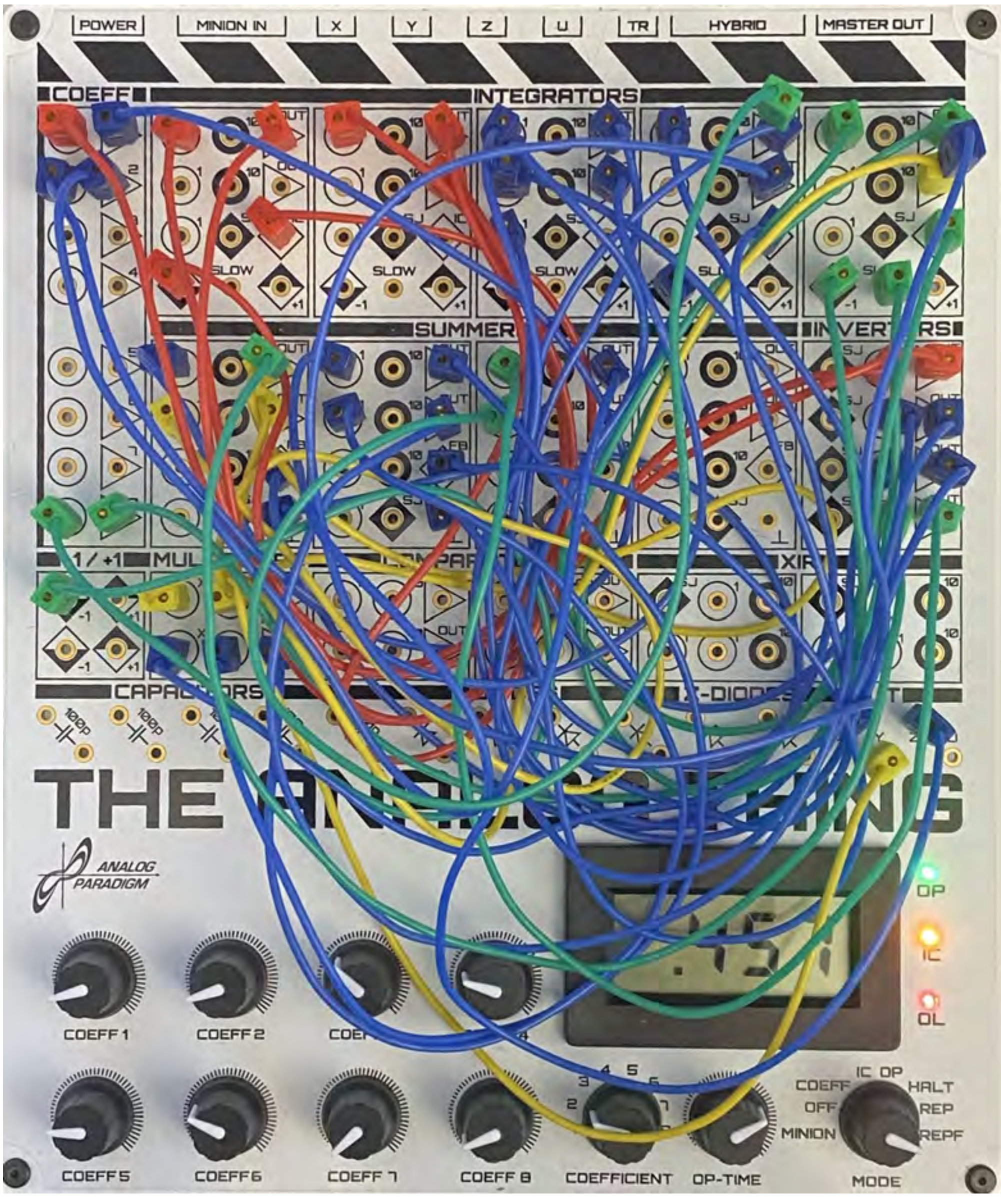

Figure 7: Setup of both approaches to computing $\mathrm{sinc}(t)$ on THE ANALOG THING

Happy analog computing!

References

[Ulmann 2023] Bernd Ulmann, *Analog and Hybrid Computer Programming*, 2nd edition, DeGruyter, 2023