Scientific computing for geophysical problems

2. Digital signal processing

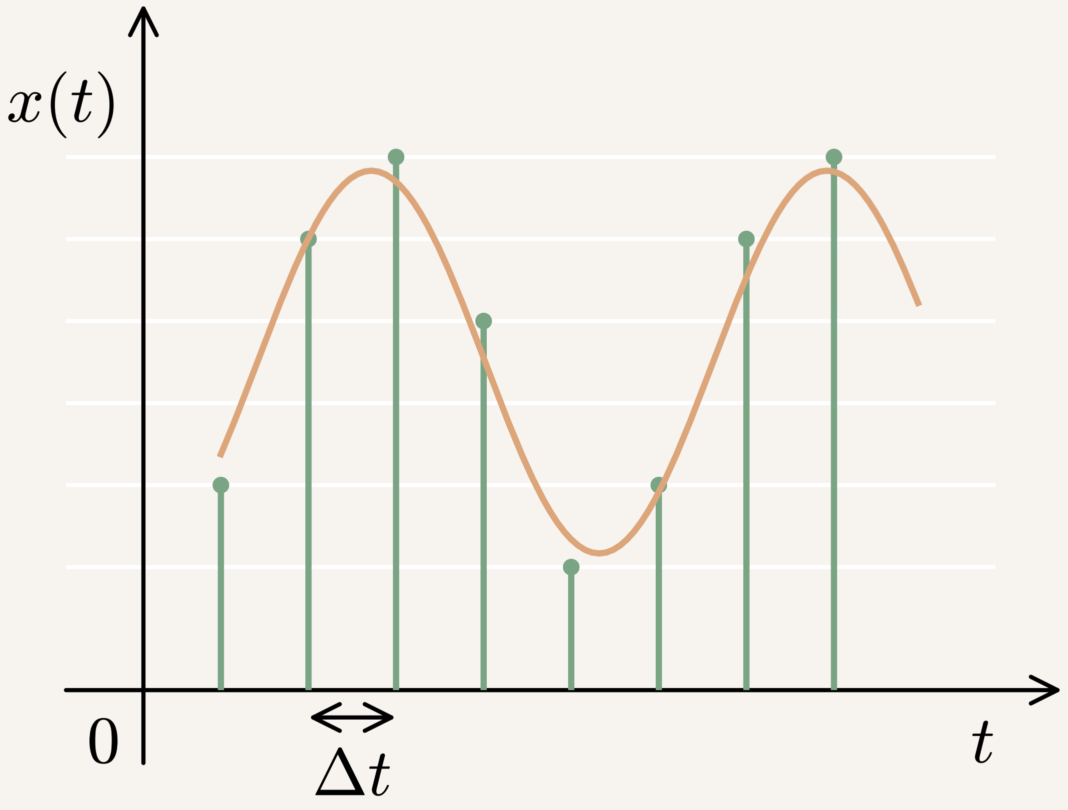
`Léonard Seydoux`

September 2024 at the institut de physique du globe de Paris.

leonard-seydoux/scientific-computing-for-geophysical-problems

Analog and digital signals

- Analog signals $x(t)$ are continuous in time and amplitude
- Digital signals $x[n]$ are discrete in time and amplitude (sampling rate, vertical resolution)



Digital signal processing in music

24 bits

0:00 / 0:05

correspond to 16,777,216 levels and 1.6MB

8 bits

0:00 / 0:05

correspond to 256 levels, but only 95kB

Signal processing as a system

A system $\mathcal{S}$ maps an input signal $x(t)$ to an output signal $y(t)$

$$x(t) \longrightarrow \boxed{\quad \mathcal{S} \quad} \longrightarrow y(t)$$

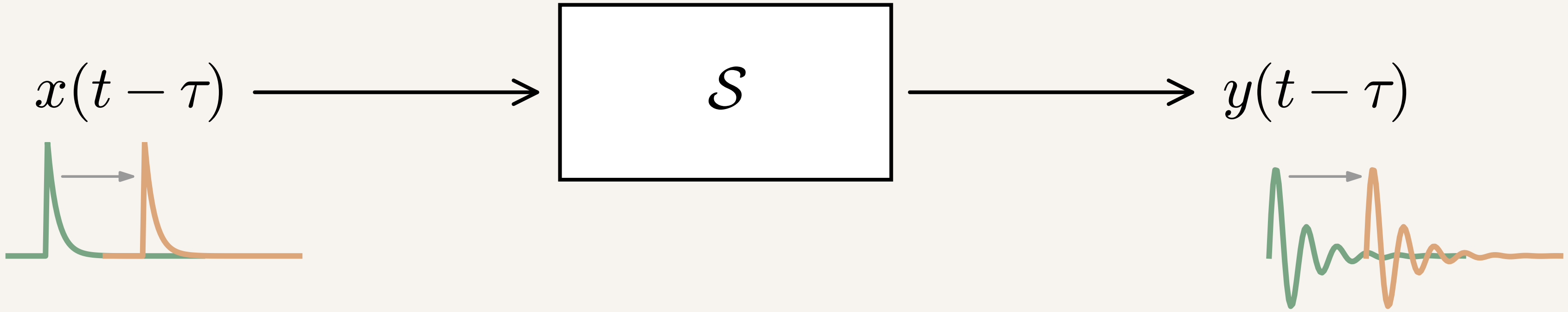
Linear systems

The system $\mathcal{S}$ is **linear** if it satisfies the following property

$$ax_1(t) + bx_2(t) \longrightarrow \boxed{\mathcal{S}} \longrightarrow ay_1(t) + by_2(t)$$

Time-invariant systems

The system $\mathcal{S}$ is **time-invariant** if it satisfies the following property



Linear and time-invariant systems

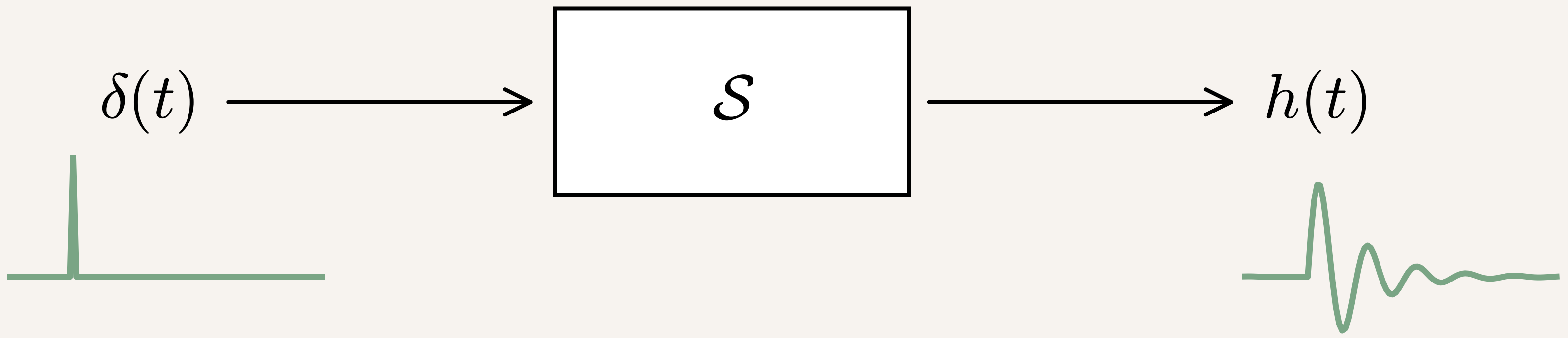
$\mathcal{S}$ is linear and time-invariant (LTI) if it satisfies both properties:

- Linearity: $\mathcal{S}\{ax_1(t) + bx_2(t)\} = ay_1(t) + by_2(t)$
- Time-invariance: $\mathcal{S}x(t-\tau) = y(t-\tau)$

Linear and time-invariant systems are fully modeled by the **impulse response**

Impulse response

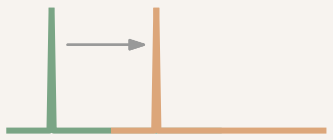
The impulse response $h(t)$ of a system $\mathcal{S}$ is the output of the system when the input is the Dirac delta function $\delta(t)$

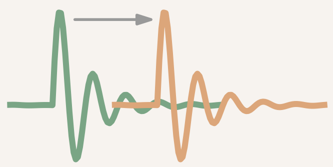


Impulse response

The impulse response $h(t)$ of a system $\mathcal{S}$ is the output of the system when the input is the Dirac delta function $\delta(t)$

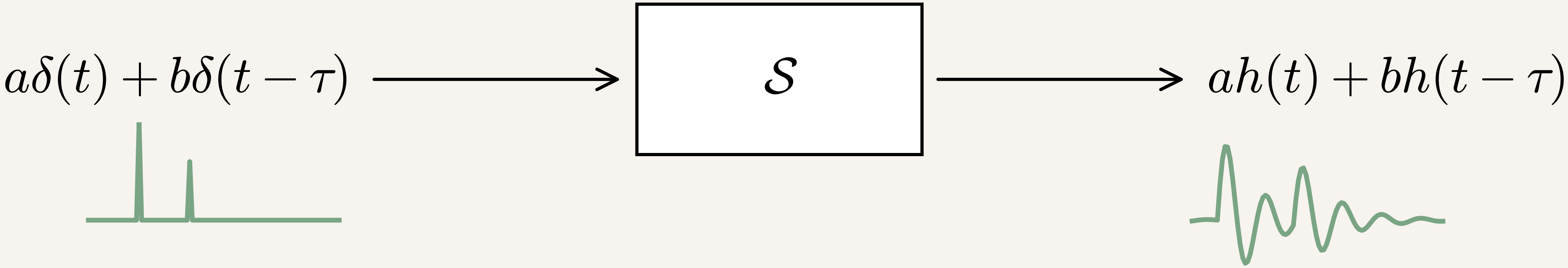
$$\delta(t-\tau) \longrightarrow \boxed{\mathcal{S}} \longrightarrow h(t-\tau)$$





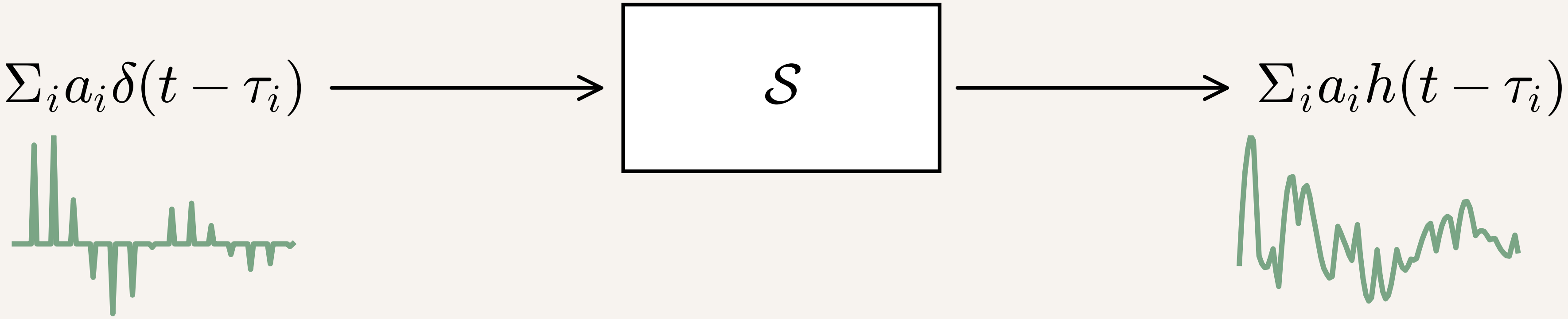
Impulse response

The impulse response $h(t)$ of a system $\mathcal{S}$ is the output of the system when the input is the Dirac delta function $\delta(t)$



Convolution

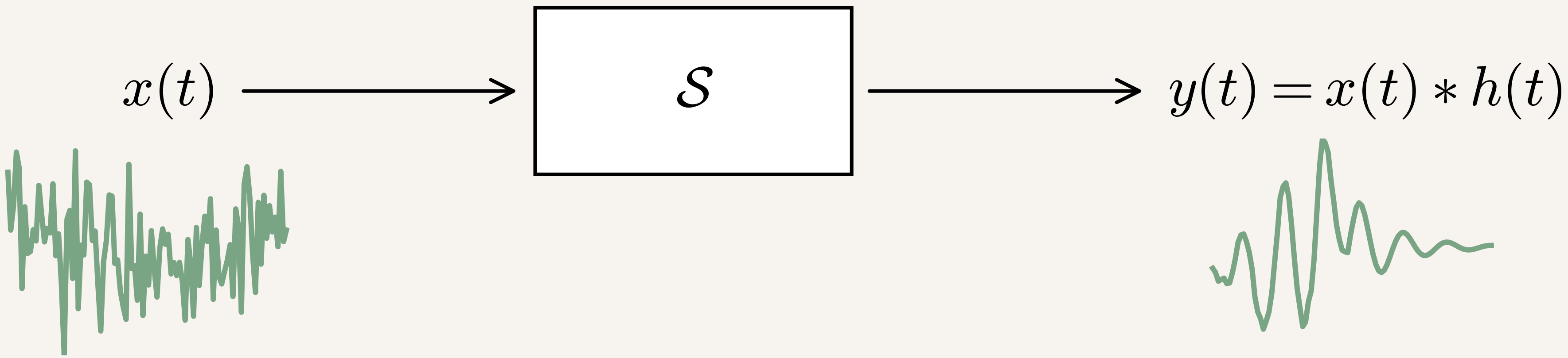
The impulse response $h(t)$ of a system $\mathcal{S}$ is the output of the system when the input is the Dirac delta function $\delta(t)$



Convolution

The output signal $y(t)$ of a system $\mathcal{S}$ is the convolution of the input signal $x(t)$ with the impulse response $h(t)$ such as

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$



Example: adding reverb to a piano record

The impulse response $i(t)$ of a large hall

0:00 / 0:07

The original piano record $x(t)$ in a dry room

0:00 / 0:05

Convolution theorem to get the piano record $y(t)$ as if it was played in the hall

$$y(t) = \frac{1}{T}\int_0^T x(\tau)i(t-\tau)d\tau$$

The piano record $y(t)$ as if it was played in the hall

0:00 / 0:13

Fourier transform (continuous)

The Fourier transform of a signal $x(t)$ is defined as

$$\hat{x}(f) = \int_{-\infty}^{+\infty} x(t)e^{-i\omega t}dt$$

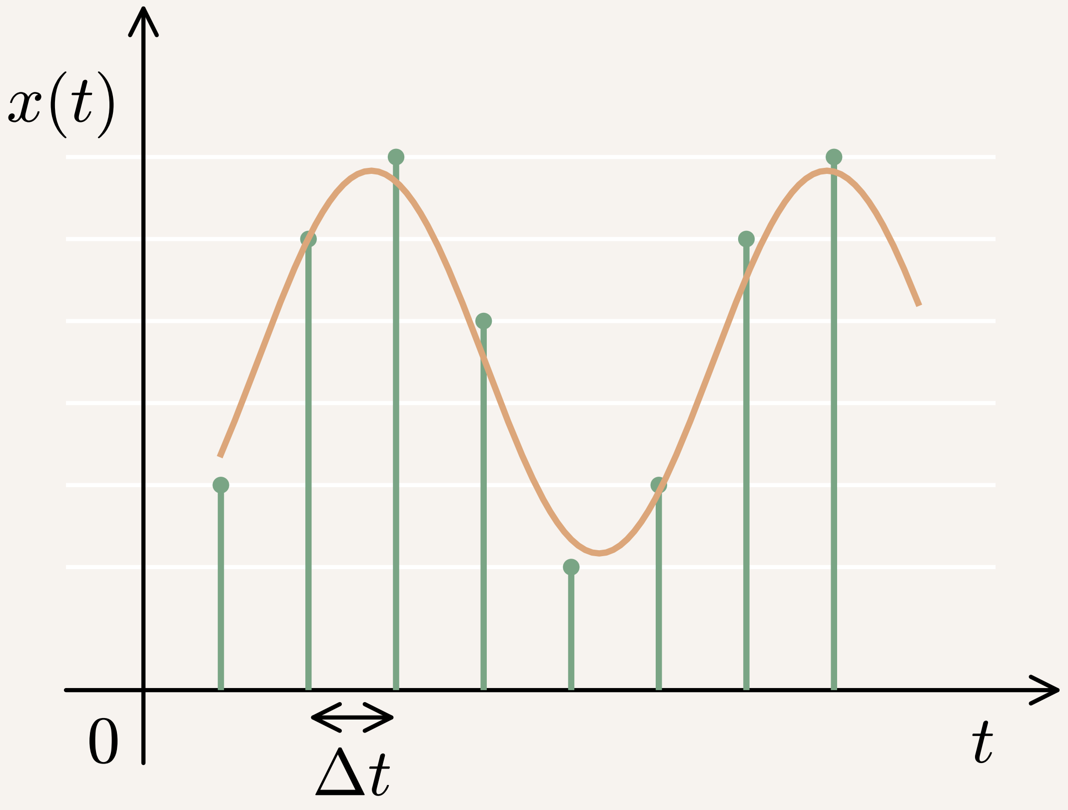
- The Fourier transform is linear: $\mathcal{F}\{ax_1(t) + bx_2(t)\} = a\hat{x}_1(\omega) + b\hat{x}_2(\omega)$
- Convolution in time is multiplication in frequency: $\mathcal{F}\{x(t) * h(t)\} = \hat{x}(\omega)\hat{h}(\omega)$

Sampling

The sampling of a continuous signal $x(t)$ at a every period Δt is defined as

$$x[n] = x(t)Ш_{\Delta t}(t)$$

where the sampling rate is defined as $f_s = \Delta t^{-1}$.



Discrete-time Fourier transform

The Discrete-time Fourier transform (DTFT) of a discrete signal $x[n]$ is defined as

$$\hat{x}(\omega) = \mathcal{F}x[n] = \sum_{n=-\infty}^{+\infty} x[n]e^{-i\omega n}$$

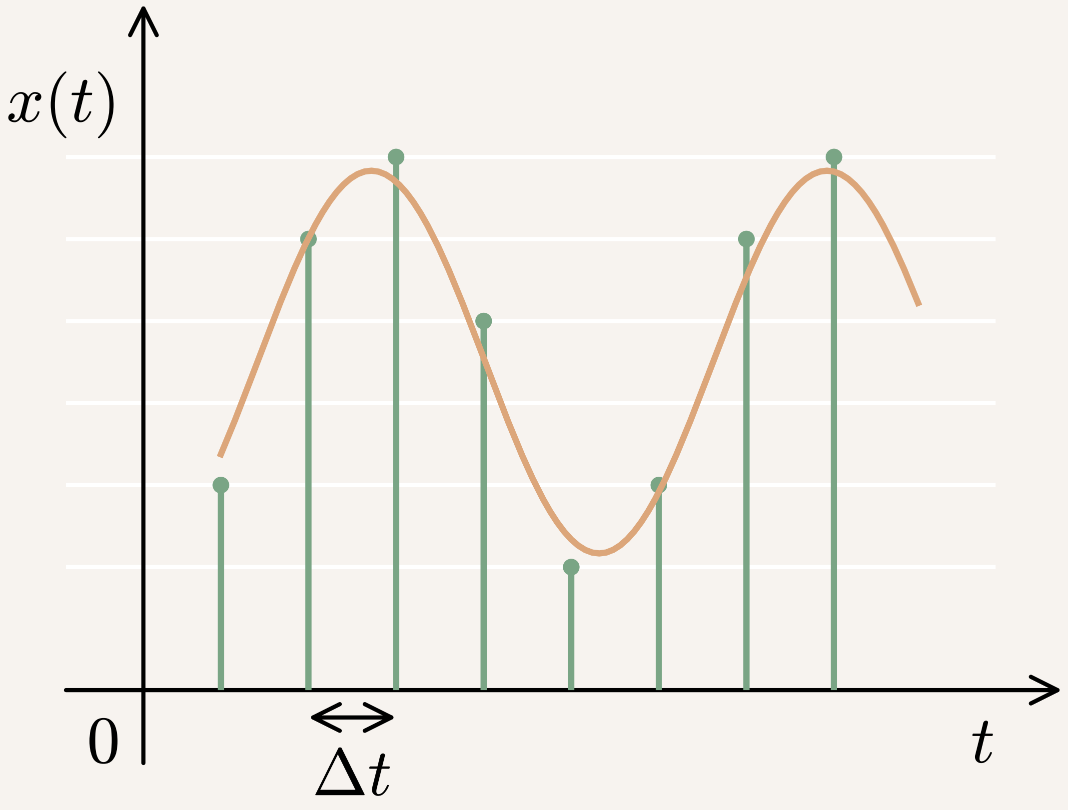
- The DTFT is linear: $\mathcal{F}\{ax_1 + bx_2\}(\omega) = a\hat{x}_1(\omega) + b\hat{x}_2(\omega)$
- Convolution in time is multiplication in frequency: $\mathcal{F}\{x * h\}(\omega) = \hat{x}(\omega)\hat{h}(\omega)$

Sampling and aliasing

In the spectral domain, sampling duplicates the spectrum at every multiple of the sampling rate ω_s.

$$\hat{x}(\omega) = \sum_{n=-\infty}^{+\infty} x(\omega - n\omega_s)$$

Sampling theorem: a signal can be reconstructed if $\omega_s \geq 2\omega_{max}$



jupyter