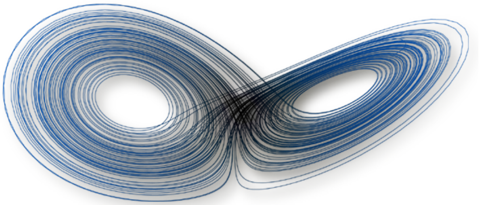


Nonlinear physics, dynamical systems and chaos theory

Jean-Christophe Loiseau

jean-christophe.loiseau@ensam.eu
DynFluid,
Arts et Métiers ParisTech, France

First-order systems

Flows on the line

First-order systems

An apparently simple system

- Let us consider the following first-order dynamical system

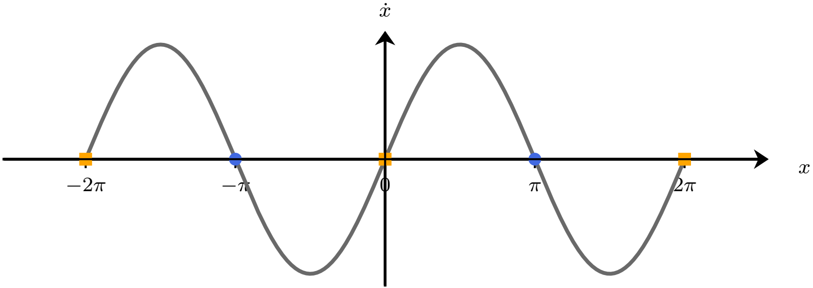
$$\dot{x} = \sin(x).$$

- Its analytical solution is given by

$$t = \ln \left| \frac{\csc(x_0) + \cot(x_0)}{\csc(x) + \cot(x)} \right|.$$

First-order systems

Phase line



First-order systems

Fixed points

- Fixed points x^* are equilibrium solutions characterized by

$$f(x^*) = 0.$$

- In the present case, these are given by

$$x^* = n\pi \text{ for } n \in \mathbb{N}.$$

First-order systems

Linear stability

- The dynamics of a perturbation $\eta(t) = x(t) - x^*$ is given by

$$\dot{\eta} = f(x^* + \eta).$$

- If η is small enough, $f(x^* + \eta)$ can be approximated by its first-order Taylor expansion around x^*

$$f(x^* + \eta) = f(x^*) + f'(x^*)\eta + \mathcal{O}(\eta^2).$$

First-order systems

Linear stability

- Given that $f(x^*) = 0$, the dynamics of η are governed by

$$\dot{\eta} = f'(x^*)\eta.$$

- Its analytical solution is given by

$$\eta(t) = \exp\left(f'(x^*)t\right)\eta_0.$$

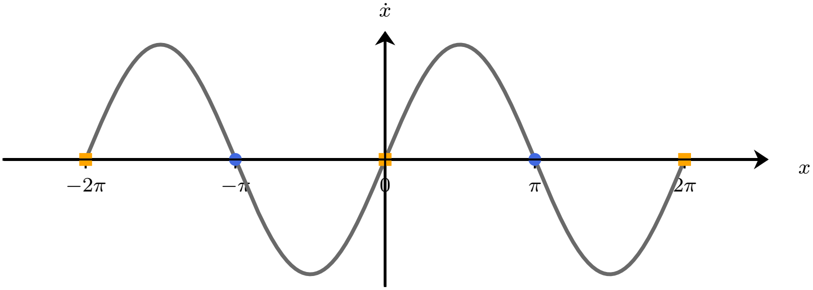
First-order systems

Linear stability

- The linear stability of a fixed point x^* is determined by the sign of $f'(x^*)$:
  - $\hookrightarrow$ if $f'(x^*) > 0$, $\eta(t)$ growths exponentially fast. The fixed point is said to be **linearly unstable**.
  - $\hookrightarrow$ if $f'(x^*) < 0$, $\eta(t)$ decays exponentially fast. The fixed point is said to be **linearly stable**.
  - $\hookrightarrow$ if $f'(x^*) = 0$, one can not conclude and nonlinear analyses are required.
- Let us now re-analyze our original system and sketch the evolution of $x(t)$.

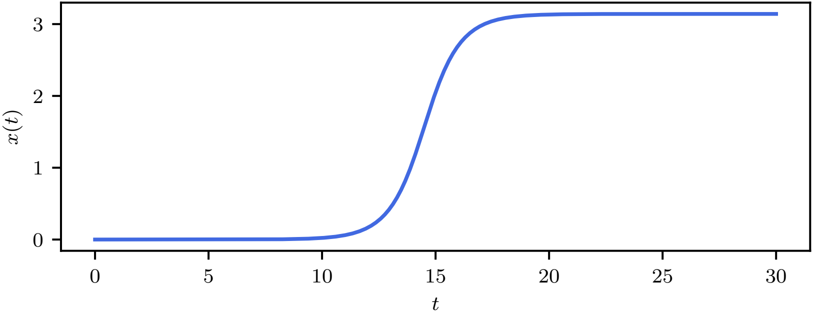
First-order systems

Phase line



First-order systems

Evolution of $x(t)$



Second-order systems

Oscillators, but not only...

Interlude

How to compute fixed points?

How to compute fixed points?

Different techniques

- Fixed points are structuring the phase space of the dynamical system under scrutiny. Unfortunately, it may not be easy (nor possible) to compute them analytically.

- A number of different numerical techniques exist for that purpose. The following list is by no means exhaustive:
  - ↪ Newton-Raphson method,
  - ↪ Selective Frequency Damping,
  - ↪ BoostConv,
  - ↪ ...

Newton-Raphson method

$f : \mathbb{R} \to \mathbb{R}$

Newton-Raphson method

$f : \mathbb{R} \to \mathbb{R}$

- After k iterations, the basic iteration scheme can be written as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

- Iteration stops when a user-defined criterion is fulfilled, usually

$$\|f(x_k)\| \leq \epsilon \text{ or } \|x_{k+1} - x_k\| \leq \epsilon,$$

with $\epsilon \simeq 10^{-10}$.

Newton-Raphson method

$f : \mathbb{R} \to \mathbb{R}$

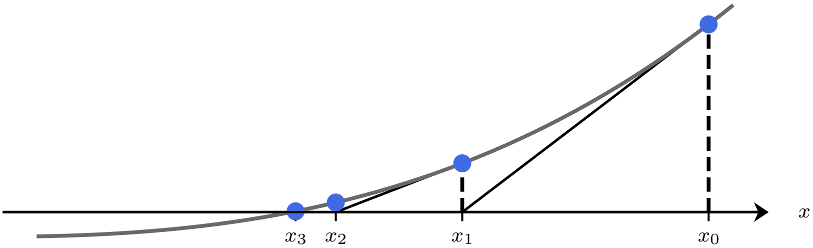


Illustration of the Newton-Raphson for $f(x) = x^3 - 2x - 5$.

Newton-Raphson method

$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$

- Generalization of the Newton-Raphson method to the case $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is quite straightforward.
- Given an estimate $\boldsymbol{x}_k$, the basic iteration reads

$$
\begin{aligned}
\boldsymbol{J} \delta \boldsymbol{x} &= -\boldsymbol{f}(\boldsymbol{x}_k) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \delta \boldsymbol{x},
\end{aligned}
$$

where $\boldsymbol{J}$ is the Jacobian matrix of $\boldsymbol{f}$ evaluated at $\boldsymbol{x}_k$.

Newton-Raphson method

$\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Newton-Raphson method

Limitations

- Although efficient, Newton-Raphson method suffers from a number of limitations:
  - ↪ The fixed points computed may depend on the initial guess $\boldsymbol{x}_0$.
  - ↪ Evaluating $\boldsymbol{f}(\boldsymbol{x})$ might be computationally expensive.
  - ↪ At each iteration, the Jacobian matrix $\boldsymbol{J}$ needs to be evaluated and inverted ($\mathcal{O}(n^3)$ operations).

- A number of variants of the Newton-Raphson method exist as to address these different limitations. This however is algorithmic refinement beyond the scope of the present course.