Systems Laboratory, Spring 2025

Damiano Varagnolo – CC-BY-4.0

notes

- welcome to the course!
- on this side of this document you will find notes that accompany the text typically visualized in class
- these notes are meant to convey the messages that are not displayed in the text on the side, and basically constitute what the teacher intends to say in class

Table of Contents I

notes

- this is the table of contents of this document; each section corresponds to a specific part of the course

Is this function a solution of this ODE?

notes

Contents map

developed content units	taxonomy levels
ODE	u1, e1

prerequisite content units	taxonomy levels
derivative	u1, e1

notes

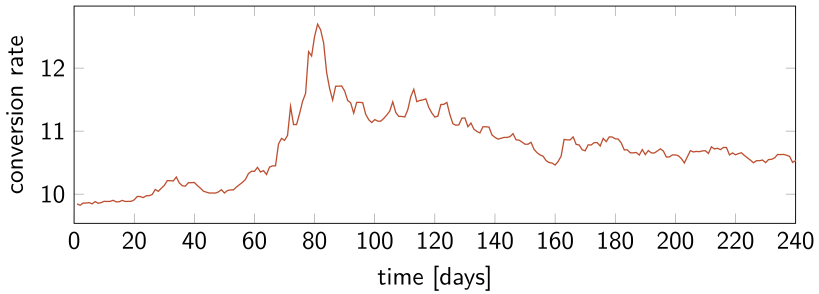
Main ILO of sub-module "Is this function a solution of this ODE?"

Decide whether a given function is a solution to a specified ODE by direct verification

notes

- by the end of this module you shall be able to do this

What is a signal?

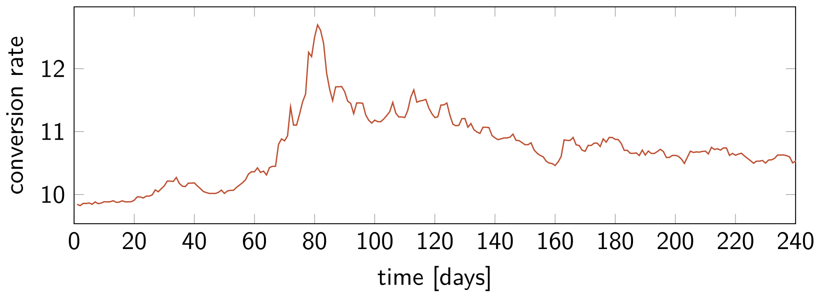


$y(t)$ (assuming t continuous in this module)

notes

- a signal is any time-dependent (or space-dependent) variation of a physical quantity that conveys information. In engineering and science, signals can take many forms, such as electrical voltages, sound waves, images, or sequences of numbers in a digital system

What is the derivative of this signal?

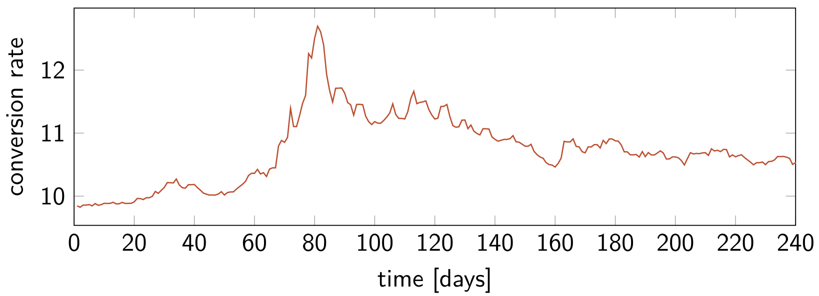


$$y(t) \quad \mapsto \quad \frac{dy(t)}{dt} = \dot{y}(t)$$

notes

- we will extensively use this notation in the course; basically never use the 'd / dt'

Would you say that $y(t) = \dot{y}(t)$, in this case?

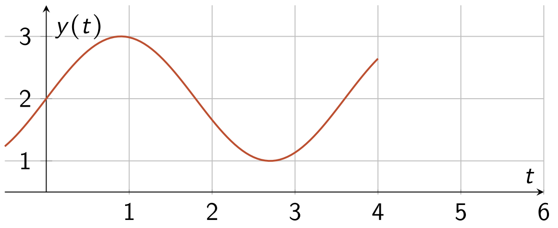


notes

- definitely not; the two signals are not the same at all *"uhm, where are we going with all this stuff?"*

$\mapsto$ be able to do forecasts

would you be able to compute $y(5)$ from this graph, if you knew that $\dot{y}(t) = y(t)$?



notes

- we will see how an ODE (Ordinary Differential Equation) describes how a system's state evolves over time based on its current state and inputs. By the end of the course it will be obvious how solving the ODE with given initial conditions, we can predict future states. This is used in control to enable forecasting, that is essentially by propagating system dynamics forward in time. We will also see though that the accuracy of forecasts depends on how well the ODE models the real-world system (something that may be more or less good)

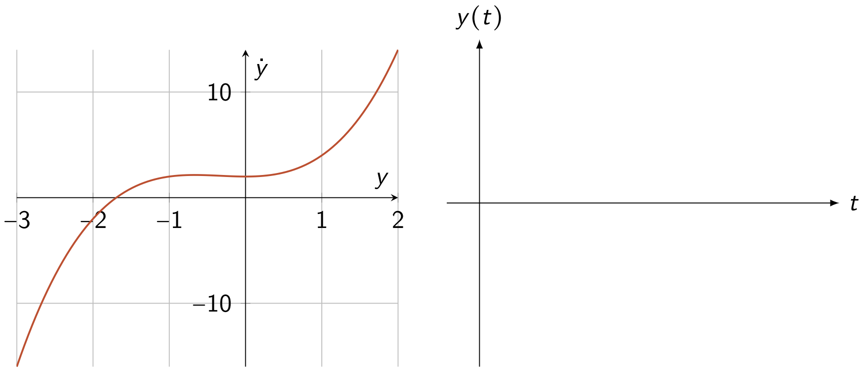
an ODE is a tool to produce forecasts

notation: instead of $\dot{y}(t)$ or $y(t)$ we will write $\dot{y}$ or y

notes

- so this is an important message
- watch out at the notation; we will always mean signals, and not scalars

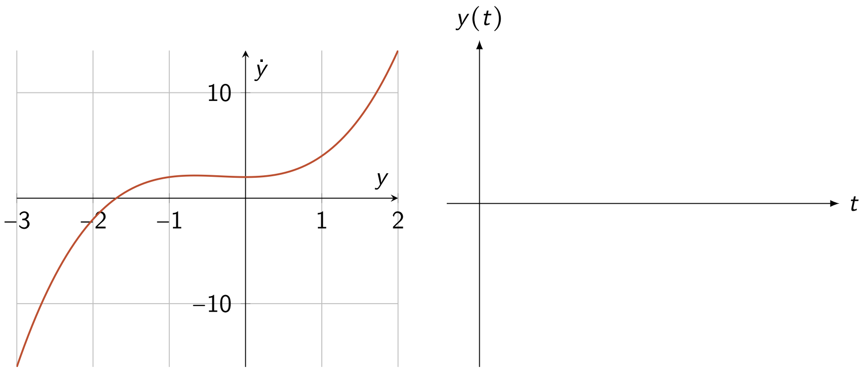
But what does it mean to solve an ODE?



notes

- before continuing we need though to reflect on what it means to solve a differential equation. Let's take this example to investigate this concept
- remember that all the symbols here actually mean signals in time, i.e., they should be intended as $\dot{y}(t) = y(t) + u(t)$ ideally with t in a certain domain (typically $[0, +\infty)$)
- for example this combination of signals does not satisfy the equation
- this instead does

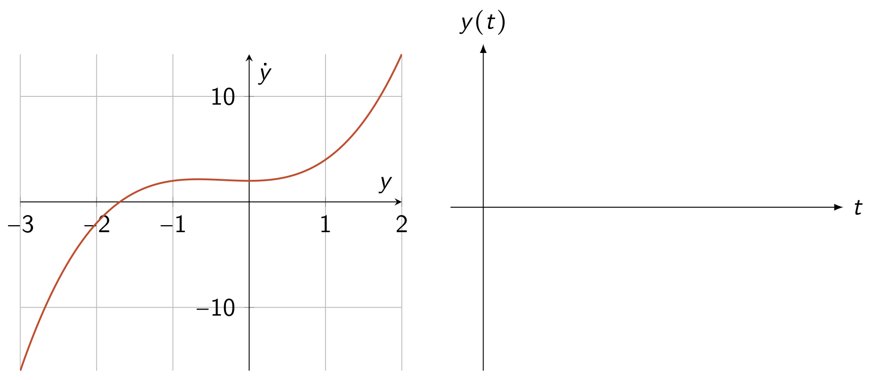
Is knowing the ODE enough to be able to generate a trajectory?



notes

- no, there is the need for defining an initial condition to be able to draw the trajectory

Does $\{y(t) = \cos(t), y(0) = 1\}$ solve this ODE?



notes

- no, because that specific signal does not satisfy the ODE

Are we done with this?

Decide whether a given function is a solution to a specified ODE by direct verification

→ no, there are still a lot of cases we shall cover

notes

- by the end of this module you shall be able to do this

Notation time!

In control, modelling a dynamical system = defining

$$\dot{\boldsymbol{y}} = \boldsymbol{f}\left(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{d}, \boldsymbol{\theta}\right),$$

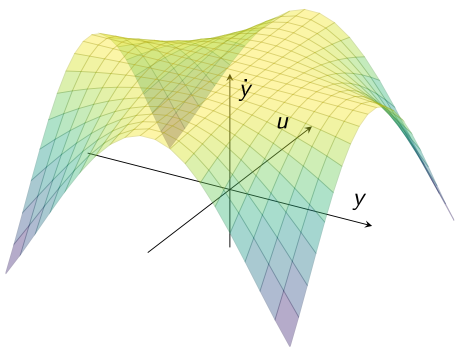
thus defining:

- the variables
  - $\boldsymbol{u}$ = inputs *(i.e., what we can steer)*
  - $\boldsymbol{d}$ = disturbances *(i.e., what we cannot steer but that still influences the system)*
  - $\boldsymbol{y}$ = outputs *(i.e., what we are interested into)*
- the shape of $\boldsymbol{f}$
- the value of its parameters $\boldsymbol{\theta}$
- bold font = vector

notes

- let's now generalize the ODE before to something that can be applied to more cases
- let's do this definition, where the names are given in this way for historical reasons
- for example the Lotka Volterra model that we will see below is a specific example of this way of writing things, where there is no u and d by the way
- once again $\boldsymbol{f}$ has the meaning of indicating "where" the system is going towards, in time - also this will be more clear soon

A graphical example of $\dot{y} = f(y, u)$



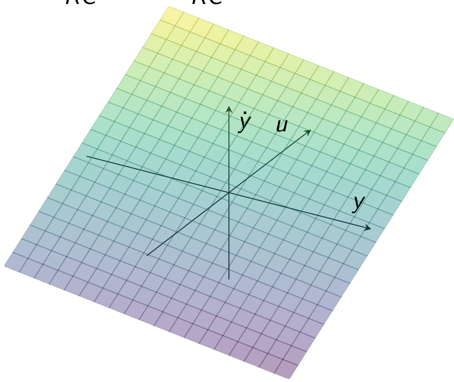
https://www.geogebra.org/classic/mmppe6hs

notes

- the simplest trajectories are that ones that seem simple constant numbers

A couple of ODEs that you may have already seen, of the type $\dot{y} = ay + bu$

- velocity of a cart: $\dot{v}(t) = -\frac{k}{m}v(t) + \frac{k}{m}F(t)$
- RC-circuits: $\dot{v}_C(t) = -\frac{1}{RC}v_C(t) + \frac{1}{RC}V(t)$



notes

- you should have already seen these ODEs (or be able to derive them) from physics and electronics

Some more details about the first example

$$\text{notation: } F(t) = ma(t) = m\dot{v}(t) \quad \mapsto \quad F = ma = m\dot{v} \tag{1}$$

...but $v = \dot{p}$, thus:

$$\begin{cases} \dot{p} = v \\ \dot{v} = \frac{F}{m} \end{cases} \tag{2}$$

$\boxed{\text{this is a system of ODEs}}$

notes

- first of all we will virtually always drop the 't' because we know we are working with time signals
- moreover we also know that velocity is a time derivative of position, so we actually obtain a system of ODEs from Newton laws
- in the next slide we see what is the geometric meaning of an ordinary differential equation or systems of ODEs

Another practical example

temperature of a small brick in a very large room whose temperature is 20 degrees:

$$\dot{T} = -0.5\left(T - 20\right)$$

notes

- let's do a practical example of a very simple ODE
- here we can see that starting from any T we ideally tend to go to T_a
- but how? Let's see

notes

- https://en.wikipedia.org/wiki/The_Treachery_of_Images

Important point: model ≠ real world

Ceci n'est pas une brique.

$$\dot{T} = -0.5\big(T - 20\big)$$

notes

- we will re-consider this point later on, but for now it has to be obvious that this is an approximated description of reality
- why we use them and what are the usages we'll do about this, we'll see later on in the course

Another example: a Lotka-Volterra model (≠ real world):

$$\begin{cases} \dot{y}_{\text{rabbits}} & = 0.4 \cdot y_{\text{rabbits}} - 0.5 \cdot y_{\text{rabbits}} \cdot y_{\text{foxes}} \\ \dot{y}_{\text{foxes}} & = -3 \cdot y_{\text{foxes}} + 0.7 \cdot y_{\text{rabbits}} \cdot y_{\text{foxes}} \end{cases}$$

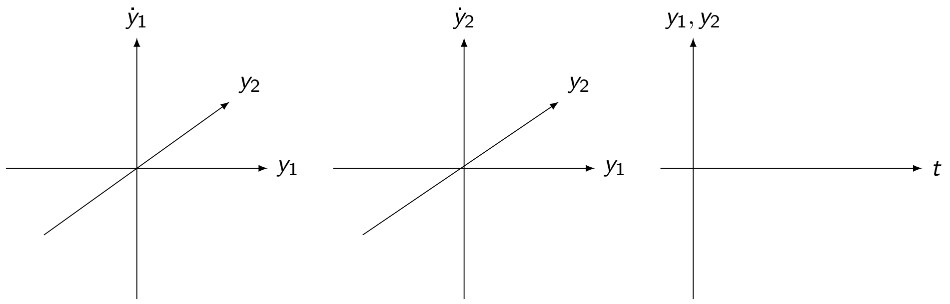
notes

- we may have though more complicated models
- for example this one is a model that may accurately describe the evolution of the populations of preys and predators in a closed environment (we'll discuss this one better later on)

What do we mean with “dynamics”? More geometrically

(example: 2D system, autonomous)

example: two dimensional $\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y})$ in the sense of $\begin{cases} \dot{y}_1 &= f_1(y_1, y_2) \\ \dot{y}_2 &= f_2(y_1, y_2) \end{cases}$

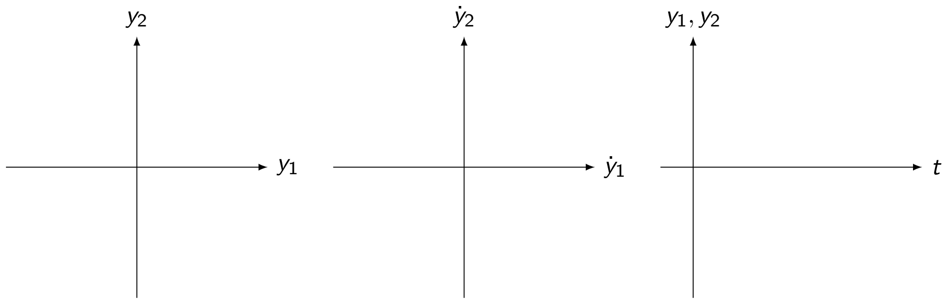


notes

- graphically we have different objects
- one typically looks at trajectories in time
- but one may also think at the phase portrait of the outputs (or states, if one is working with state space models)
- the important is to realize that for any given couple $\boldsymbol{y}, \boldsymbol{u}$ one is able to compute f, that means being able to compute the vector $\dot{\boldsymbol{y}}$, that can be seen in the first set of cartesian axes
- then this $\dot{\boldsymbol{y}}$ can also be superposed in the axes for $\boldsymbol{y}$, and we can give the interpretation that this is where the system is moving towards
- intuitively, changing u means changing $\dot{\boldsymbol{y}}$, that means also steering where the system is heading towards
- note that the fact that they system has some dynamics means that if I want to reach a specific $\boldsymbol{y}$ starting from a given initial condition $\boldsymbol{y}_0$, there is the need for waiting a bit of time before we can reach the goal. We cannot go 'instantaneously' to the goal

Same example, alternative viewpoint

$$\begin{cases} \dot{y}_1 &= f_1(y_1, y_2) \\ \dot{y}_2 &= f_2(y_1, y_2) \end{cases}$$



notes

- we may though see this with an alternative viewpoint, that will lead us to linear algebra formulations

Coding the Lotka-Volterra example

$$\begin{cases} \dot{y}_{\text{prey}} &= \alpha y_{\text{prey}} - \beta y_{\text{prey}} y_{\text{pred}} \\ \dot{y}_{\text{pred}} &= -\gamma y_{\text{pred}} + \delta y_{\text{prey}} y_{\text{pred}} \end{cases}$$

`./LotkaVolterraSimulator.ipynb`

(we'll see later on how this continuous thing is actually implemented in our discrete computers)

notes

- to solve numerically the trajectories let's explore and run the corresponding python notebook
- for now we will do things numerically; later on we will do it also analytically

Discussion: did we model the Lotka-Volterra dynamical system here?

```
def myModel(y, t):
 #
 # parameters
 alpha = 1.1
 beta  = 0.4
 gamma = 0.4
 delta = 0.1
 #
 # get the individual variables - for readability
 yPrey = y[0]
 yPred = y[1]
 #
 # individual derivatives
 dyPreydt  =   alpha * yPrey - beta  * yPrey * yPred
 dyPreddt  = - gamma * yPred + delta * yPrey * yPred
 #
 return [ dyPreydt, dyPreddt ]
```


notes

- do you think that this code represents a model? yes, this contains all the information that is needed to compute the f, i.e., the direction where the system is pointing towards given the current situation of the system

Discussion: do we need something more than just the model to simulate the system?

$$\begin{cases} \dot{y}_{\text{prey}} &= 1.2 y_{\text{prey}} - 0.1 y_{\text{prey}} y_{\text{pred}} \\ \dot{y}_{\text{pred}} &= -0.6 y_{\text{pred}} + 0.2 y_{\text{prey}} y_{\text{pred}} \end{cases}$$

notes

- do you think we can simulate the model before just with this information? Or do we need something more? no, because we need an initial condition. Starting from different initial conditions we have completely different behaviors (think at no foxes vs there are some foxes)

Remember: static ≠ dynamic

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}) \qquad \neq \qquad \dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\theta})$$

notes

- thus remember always that we are working with ODEs, that means that f does not give what is the output directly, but rather where the system is going to. It gives a derivative!

Summarizing

Decide whether a given function is a solution to a specified ODE by direct verification

- check y, compute $f(y)$, compute $\dot{y}$
- does $f(y) = \dot{y}$?
- same apply for higher orders / more complex ODES from notational perspectives

notes

- you should now be able to do this

Most important python code for this sub-module

notes

Solving ODEs

https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter22.06-Python-ODE-Solvers.html

notes

- check that page

Self-assessment material

notes

Question 1

Which of the following best describes what it means for a function $y(t)$ to be a solution of an ODE?

Potential answers:

I: **(wrong)** It satisfies the ODE for at least one value of t.
II: **(correct)** It satisfies the ODE for all values of t in its domain.
III: **(wrong)** It approximately satisfies the ODE within a certain error margin.
IV: **(wrong)** It satisfies the ODE only at integer values of t.
V: **(wrong)** I do not know

Solution 1:

A function is a solution of an ODE if it satisfies the equation for all values of t within its domain. A solution must be valid throughout the considered interval, not just at isolated points.

notes

- see the associated solution(s), if compiled with that ones :)

Question 2

What additional information is needed to uniquely determine a solution of an ODE?

Potential answers:

I: **(wrong)** The function $y(t)$ itself.
II: **(correct)** An initial condition specifying the value of y at a given time.
III: **(wrong)** A boundary condition at two different points.
IV: **(wrong)** The highest-order derivative of y.
V: **(wrong)** I do not know

Solution 1:

An initial condition provides the necessary information to select a unique solution from the family of possible solutions to a differential equation. Without it, multiple solutions may exist.

notes

- see the associated solution(s), if compiled with that ones :)

Question 3

Given the ODE $\dot{y} = y$, which of the following functions is a solution?

Potential answers:

I: **(wrong)** $y(t) = t^2$
II: **(correct)** $y(t) = Ce^t$, where C is a constant.
III: **(wrong)** $y(t) = \sin t$
IV: **(wrong)** $y(t) = \frac{1}{t+1}$
V: **(wrong)** I do not know

Solution 1:

The function $y(t) = Ce^t$ satisfies the equation since its derivative is also Ce^t, matching the right-hand side of the ODE.

notes

- see the associated solution(s), if compiled with that ones :)

Question 4

Which of the following differential equations is nonlinear?

Potential answers:

I: **(wrong)** $\dot{y} + 2y = 3$
II: **(correct)** $\dot{y} = y^2$
III: **(wrong)** $\dot{y} = 3y + 5$
IV: **(wrong)** $\dot{y} + \sin y = t$
V: **(wrong)** I do not know

Solution 1:

The equation $\dot{y} = y^2$ is nonlinear because it contains a nonlinear term (y^2), whereas a linear ODE has terms that are at most first-degree in the function and its derivatives.

notes

- see the associated solution(s), if compiled with that ones :)

Question 5

What is an equilibrium point of the ODE $\dot{y} = y(1-y)$?

Potential answers:

I: **(wrong)** $y = 2$
II: **(correct)** $y = 0$ and $y = 1$
III: **(wrong)** $y = -1$
IV: **(wrong)** $y = \frac{1}{2}$
V: **(wrong)** I do not know

Solution 1:

Equilibrium points occur where $\dot{y} = 0$, meaning $y(1-y) = 0$. This happens at $y = 0$ and $y = 1$.

notes

- see the associated solution(s), if compiled with that ones :)

Recap of sub-module "Is this function a solution of this ODE?"

- a function is a solution of an ODE if it satisfies the equation for all values in its domain
- initial conditions are necessary to uniquely determine a solution

notes

- the most important remarks from this sub-module are these ones