Scientific Computing

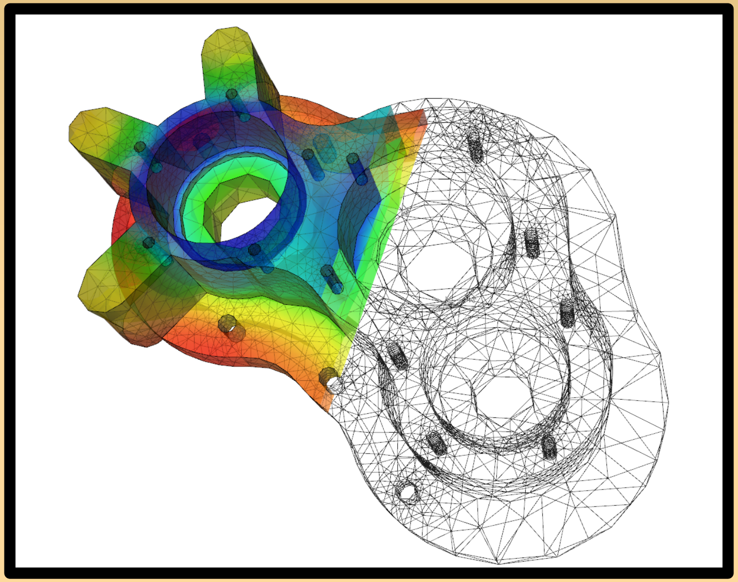
Maastricht Science Program

Week 6

Frans Oliehoek
<frans.oliehoek@maastrichtuniversity.nl>

The World is Dynamic

- Many problems studied in science are 'dynamic'
  - change over time
- Examples:
  - change of temperature
  - trajectory of a baseball
  - populations of animals
  - changes of price in stocks or options



Visualization of heat transfer in a pump casing

Heat is generated internally, cooled at the boundary → steady state temperature distribution.

- Commonly modeled with *differential equations*
  - (Not to be confused with difference equations)

Recap Difference Equations

- Remember **difference equations** (week1, week5)
  - e.g. polulation growth:

$$P_t = P_{t-1} + \Delta P_{t-1}$$

$$\Delta P_{t-1} = (b - d) P_{t-1}$$

  - discrete time steps
- Now **differential equations**: continuous time

Differential Equations

- Simple growth of bacteria model:

$$r(t) = C\,p(t)$$

  - r – rate of growth
  - p – population size

Differential Equations

- Simple growth of bacteria model:

$$r(t) = C\,p(t)$$

  - r – rate of growth
  - p – population size

Question to solve:
- How many bacteria are there at some time *t*
- given $p(t_0) = 41$
  ?

- More general: find p(t) for some range a<t<b

Differential Equations

- Simple growth of bacteria model:

$$\frac{dp(t)}{dt} = C\,p(t)$$

  - r – rate of growth
  - p – population size

This is the derivative of p!

Differential Equations

- Simple growth of bacteria model:

$$\frac{dp(t)}{dt} = C\,p(t)$$

  - r – rate of growth
  - p – population size

This is the derivative of p!

Contrast this with ΔP_{t-1}
in difference equations
→ now the change also needs to be a continuous function of time!

Differential Equations

- Simple growth of bacteria model:

$$\frac{dp(t)}{dt} = C\,p(t) \longrightarrow p'(t) = C\,p(t)$$

  - r – rate of growth
  - p – population size

Also:

$$\dot{p}(t) = C\,p(t)$$

$$\dot{p} = C\,p$$

Differential Equations

- Simple growth of bacteria model:

$$\frac{dp(t)}{dt} = C\,p(t) \longrightarrow p'(t) = C\,p(t)$$

  - r – rate of growth
  - p – population size

- Different types
  - ordinary (**ODEs**) : all derivatives w.r.t. 1 'independent variable' (vs. 'partial DE' with multiple variables)
  - Order of a DE: maximum order of differentiation.

Problem

- Given an ODE

$$y'(t)=f(t,y(t)), \qquad \forall t\in I$$

some time interval

- find a function y*(t)* that satisfies it.

Problem

$$f(t, y(t)) = C\,y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- find a function y*(t)* that satisfies it.

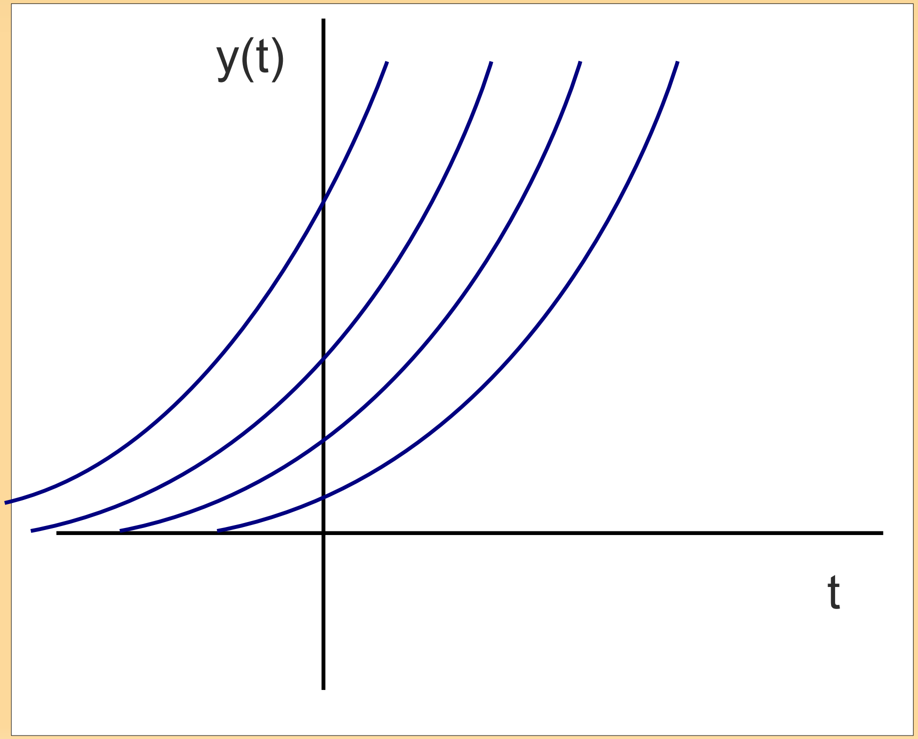
Problem

$$f(t, y(t)) = C\, y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- find a function y*(t)* that satisfies it.
- But: there are **infinitely many solutions**!



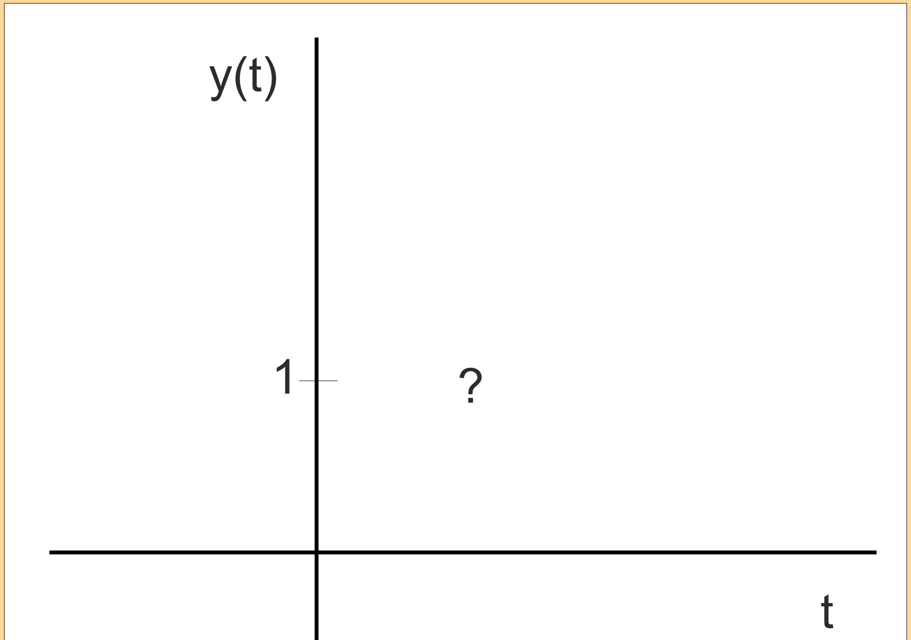
Direction Fields

$$f(t, y(t)) = C\, y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...



$$f(t, y(t)) = 1\, y(t)$$

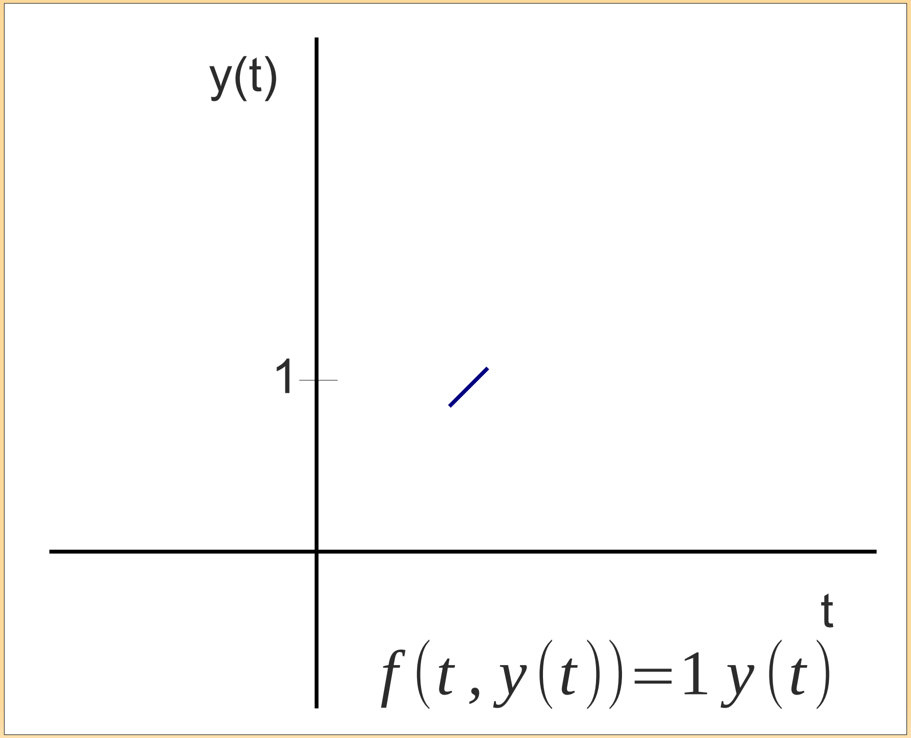
Direction Fields

$$f(t, y(t)) = C\, y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall\, t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...



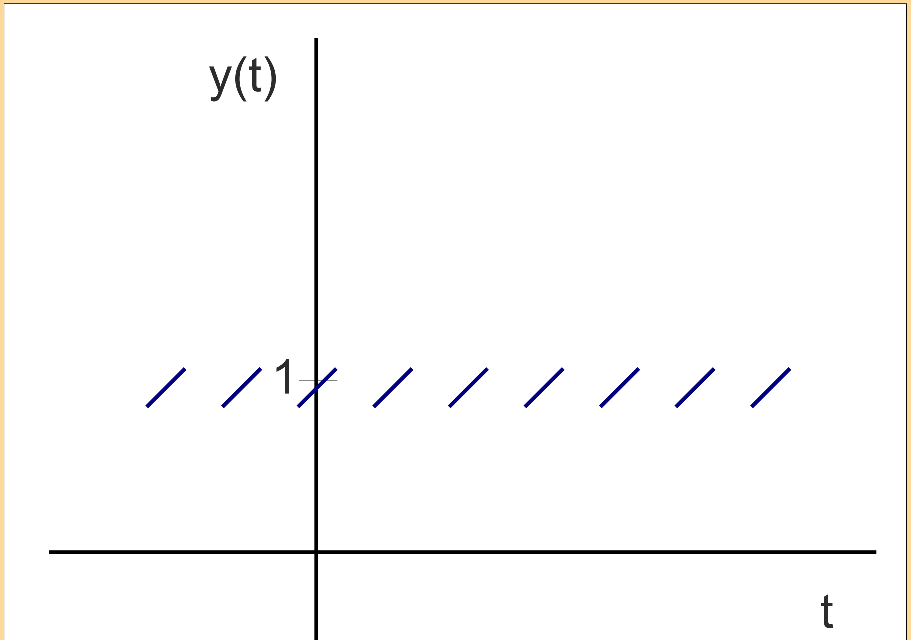
Direction Fields

$$f(t, y(t)) = C\,y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...



$$f(t, y(t)) = 1\,y(t)$$

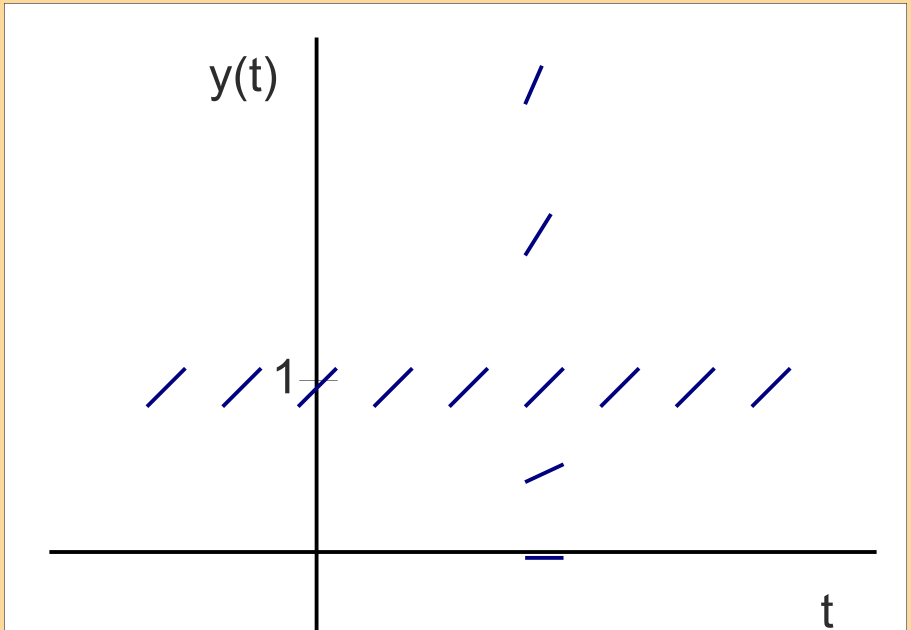
Direction Fields

$$f(t, y(t)) = C\,y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...



$$f(t, y(t)) = 1\,y(t)$$

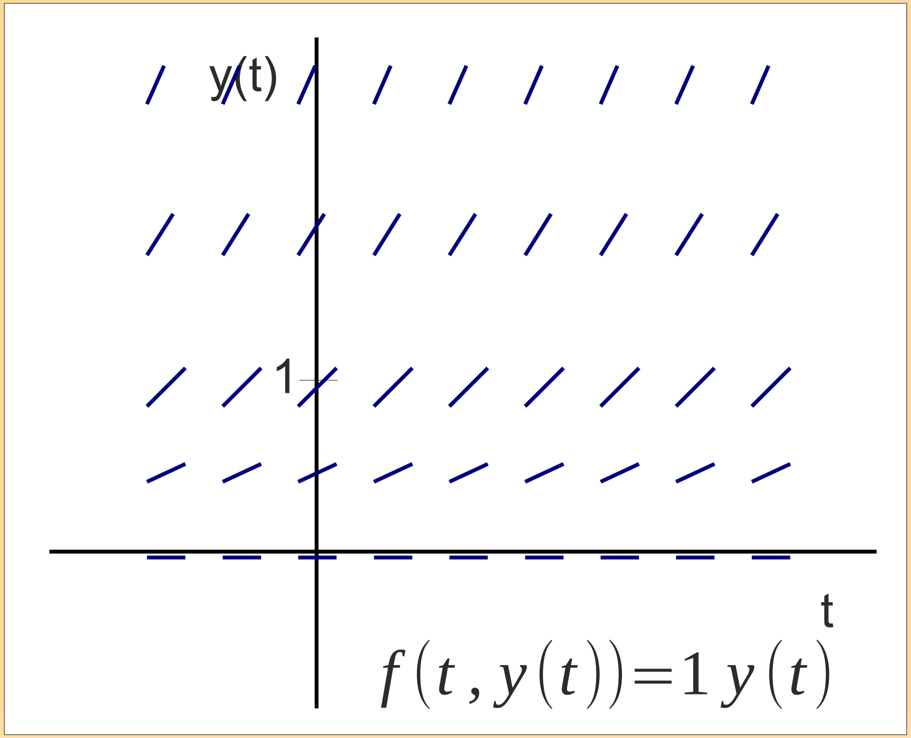
Direction Fields

$$f(t, y(t)) = C\, y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...



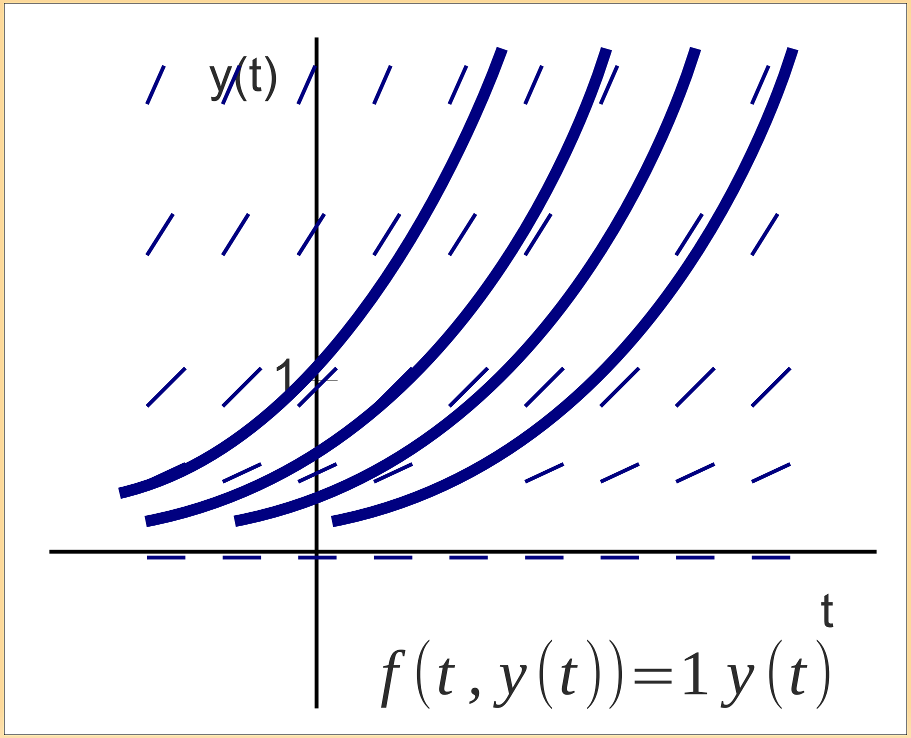
Direction Fields

$$f(t, y(t)) = C\,y(t)$$

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- Many functions satisfy it...
- Let's plot the derivatives...

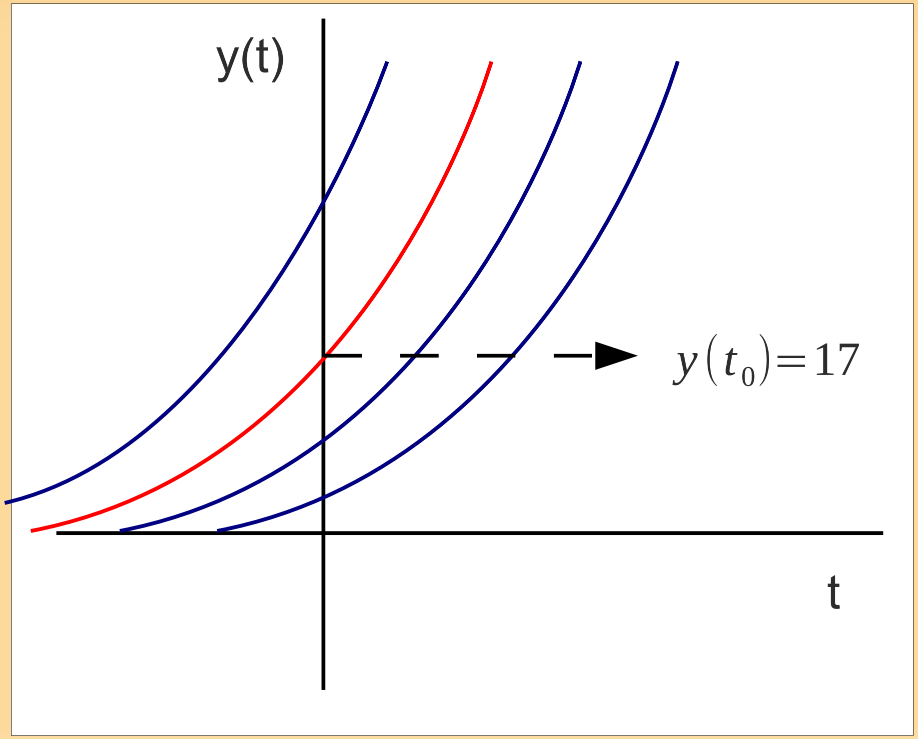


Initial Value problem

- Given an ODE

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

- find a function y*(t)* that satisfies it.
- Initial Value Problem
  (also: 'Cauchy Problem')
  - specifies $y(t_0)$
    $\longrightarrow$ unique solution

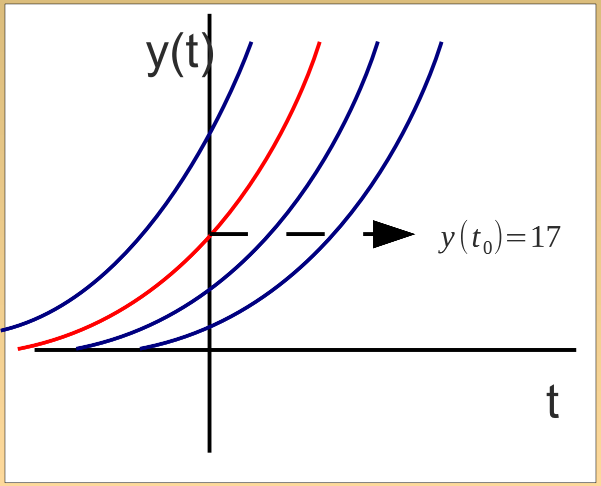


Initial Value problem

- Initial value problem:

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

$$y(t_0) = y_0$$



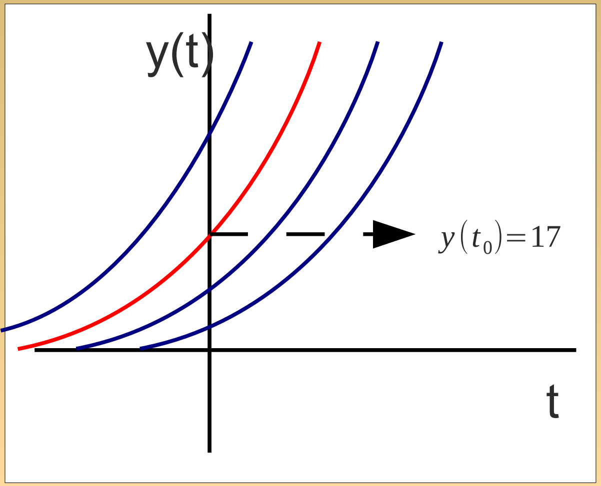
- find a function y*(t)* that satisfies it

Initial Value problem

- Initial value problem:

$$y'(t) = f(t, y(t)), \qquad \forall t \in I$$

$$y(t_0) = y_0$$



- find a function *y(t)* that satisfies it

However...

- closed-form solutions y*(t)* only available for very special cases.

  → Need for numerical solutions!

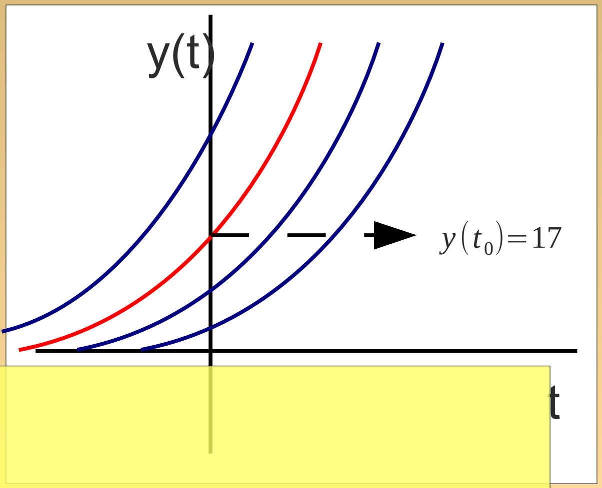
Approach

- Discretization: divide interval *I* in short steps of length *h*
- At each *node* t_n compute $u_n \approx y(t_n)$
- Numerical solution: $\{u_{0,} u_{1,} \ldots, u_N\}$

Initial Value problem

- Initial value problem:

$$y'(t)=f(t,y(t)), \qquad \forall t\in I$$
$$y(t_0)=y_0$$



- find a function *y(t)* that satisfies it

However...

- closed-form solutions y*(t)* only available for very special cases.

  → Need for numerical solutions!

Approach

- Discretization: divide interval *I* in short steps of length *h*
- At each *node* t_n compute $u_n\approx y(t_n)$
- Numerical solution: $\{u_{0,}u_{1,}\dots,u_N\}$

Effectively we perform a simulation!

Forward Euler Method

- The forward Euler method
  - just perform the 'simulation'
  - shorthand $f_n = f(t_n, u_n)$

$$u_{n+1} = u_n + h f_n$$

Forward Euler Method

- The forward Euler method
  - just perform the 'simulation'
  - shorthand $f_n = f(t_n, u_n)$

$$u_{n+1} = u_n + h f_n$$

Example

$$u_0 = 12740$$

t = (0,19)
h = 1
p(0) = 12740
r(p) = 0.1 * p

Forward Euler Method

- The forward Euler method
  - just perform the 'simulation'
  - shorthand $f_n = f(t_n, u_n)$

$$u_{n+1} = u_n + h f_n$$

Example

t = (0,19)
h = 1
p(0) = 12740
r(p) = 0.1 * p

$$u_0 = 12740$$

$$u_1 = u_0 + h * r(u_0) = 12740 + 1 * 1274.0 = 14014$$

Forward Euler Method

- The forward Euler method
  - just perform the 'simulation'
  - shorthand $f_n = f(t_n, u_n)$

$$u_{n+1} = u_n + h f_n$$

Example

t = (0,19)
h = 1
p(0) = 12740
r(p) = 0.1 * p

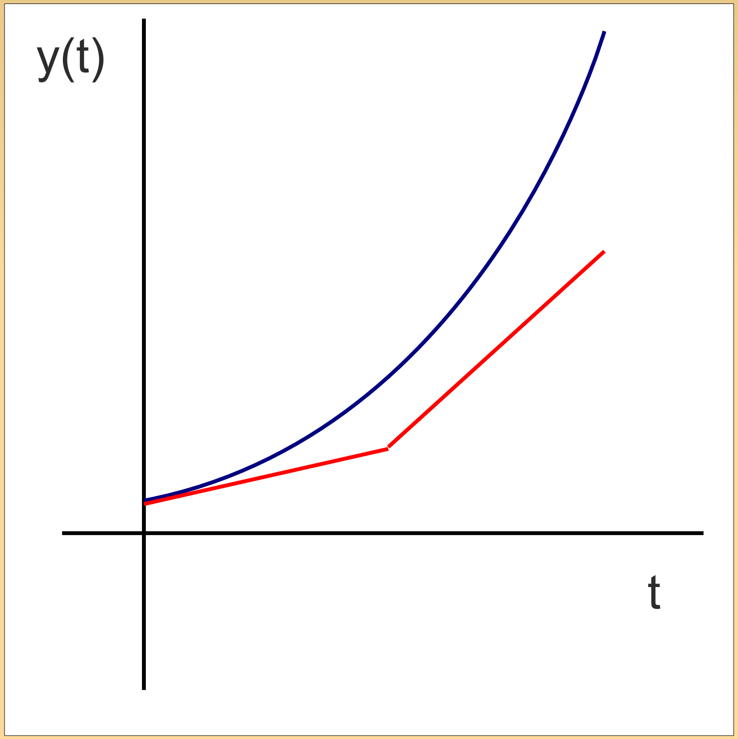
$$u_0 = 12740$$

$$u_1 = u_0 + h * r(u_0) = 12740 + 1 * 1274.0 = 14014$$

$$u_2 = u_1 + h * r(u_1) = 14014 + 1 * 1401.4 = 15415.40$$

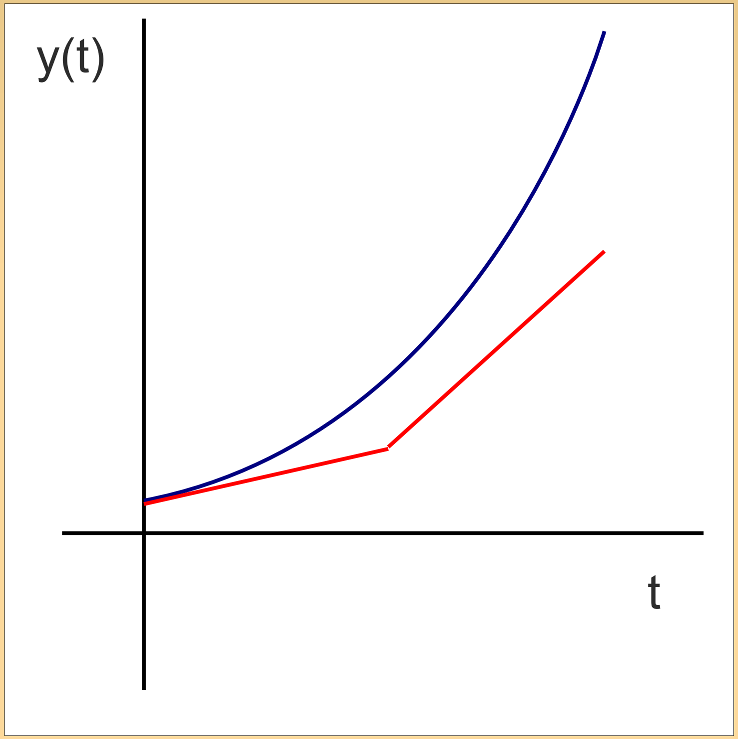
Forward Euler Method – Errors

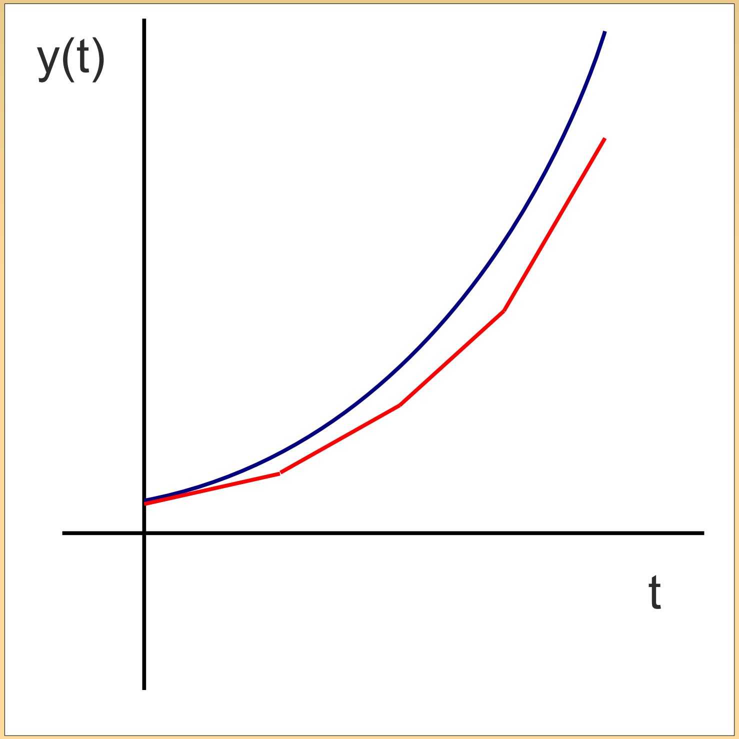
- Errors...



Forward Euler Method – Errors

- Errors...





Computational Issues

- How accurate is this?
- Does it 'converge' ?
- What is the order p of convergence?

Computational Issues

- How accurate is this?

Can we deriver an expression for the error?

- Does it 'converge' ?

if h $\rightarrow$ 0,
does error $\rightarrow$ 0 ?

- What is the order *p* of convergence?

Do we have
$$|err| < C(h) = O(h^p)$$

Computational Issues

- How accurate is this?

Can we deriver an expression for the error?

- Does it 'converge' ?

if h → 0,
does error → 0 ?

- What is the order *p* of convergence?

Do we have

$$|err| < C(h) = O(h^p)$$

- forward Euler method converges with order 1
- roughly: “h twice as small → error twice as small”
- the book discusses many methods with higher order.
- Matlab implements many:

  ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb

  - “doc ode23”

Computational Issues

- Do they matter?
  - yes...

- what to use? Matlab's doc:

  “ode45 should be first you try”