수치해석 기초

김윤수 (Caltech)

Jul 28 2022,
@ NRGW Summer School, Busan

Contents

- PDE basics

- Numerical approximation of PDE
  - numerical derivative
  - time integration

- Methods
  - Finite difference method
  - Finite volume method
  - Spectral method (*Jul 28th)

- Convergence

Partial differential equations (PDEs)

PDEs are common in physics, describing change of a system over space and time

e.g. Discharging RC circuit

$$\frac{dQ}{dt} = -\frac{1}{RC}Q$$

- unknown solution : Q → *dependent variable*
- is a function of : t → *independent variable*

e.g. 1D wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

- unknown solution : u
- is a function of : (x, t)

Differential equations with…

- 1 independent var : *ordinary differential equation (ODE)*
- $\geq$ 2 independent vars : *partial differential equation (PDE)*

Partial differential equations (PDEs)

e.g. Schrodinger equation in 1D

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi$$

PDE, 2 independent vars, 1 dependent var $\Psi(x, t)$

e.g. Lotka-Volterra equations (predator-prey model)

$$\frac{dx}{dt} = Ax - Bxy$$

$$\frac{dy}{dt} = Cxy - Dy$$

system of 2 ODEs (not PDE)

e.g. Einstein equations in vacuum

$$G_{\mu\nu} = 0$$

system of 10 PDEs,

4 independent vars, 10 dependent vars $g_{\mu\nu}(x^a)$

Partial differential equations (PDEs)

PDEs may permit analytic (exact) solutions

e.g. Wave equation (3D) $\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$ ⟶

Plane wave
Spherical wave
…

e.g. vacuum Einstein equations $G_{\mu\nu} = 0$ ⟶

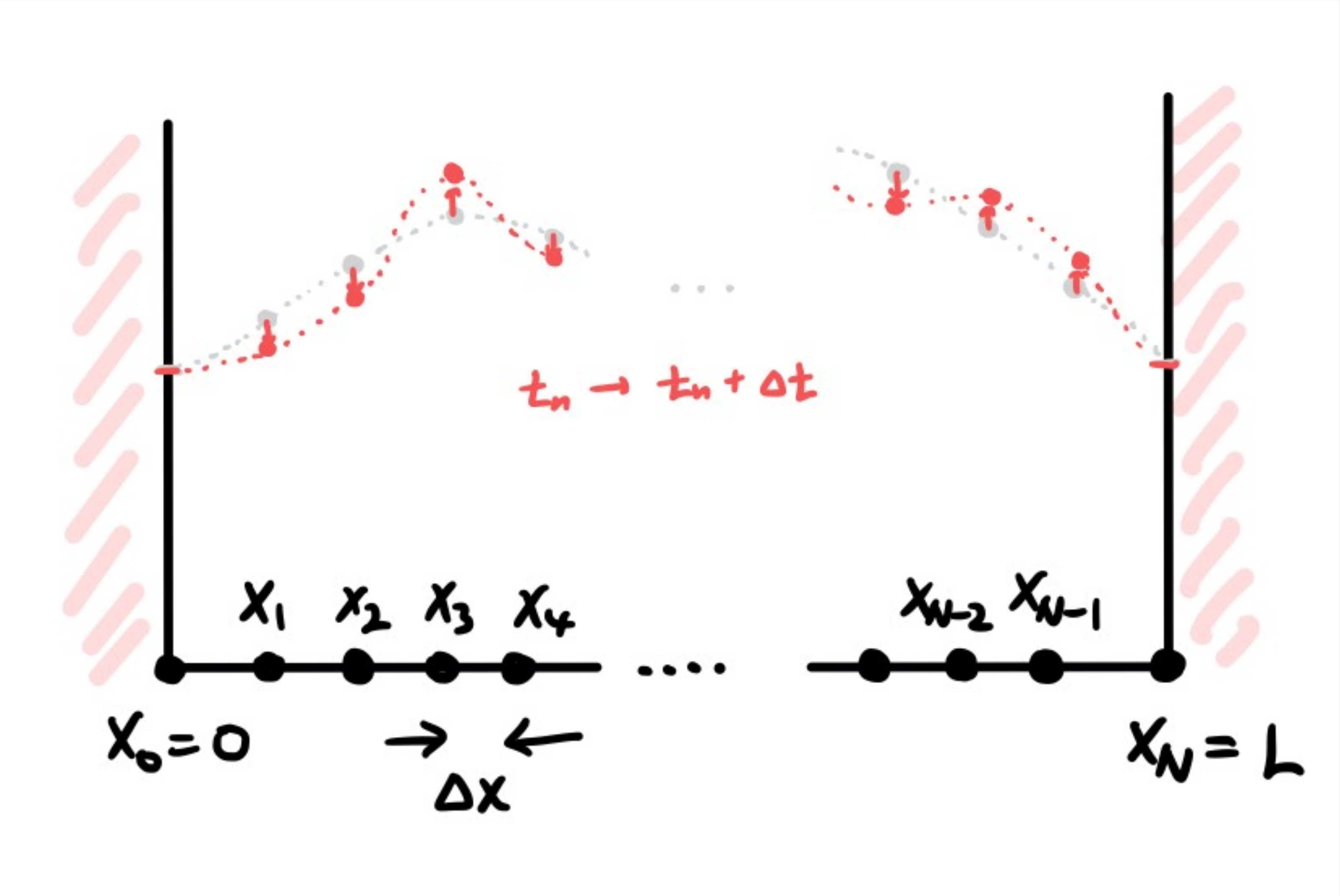
Flat (Minkowski) space
Schwarzschild geometry
…

however, in general situations, they need to be solved numerically

Example — heat equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

- $x \in [0, L]$
- $u(x,0) = u_0(x)$
- $u(0) = u(L) = 0$



Common practice :

- discretize computational domain
- approximate the PDE into discrete version
- apply initial condition
- evolve with time
- complete simulation

$$\frac{du_i}{dt} = \alpha \left(\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} \right) \equiv (*)$$

$$u_i = u_0(x_i)$$

$$u_i(t_{n+1}) = u_i(t_n) + (*) \times \Delta t$$

How to

- approximate spatial derivatives
- integrate with respect to time

Numerical derivative

How to approximate derivative on discrete grids?

$$f(x_0 + h) = f(x_0) \;\; + hf'(x_0) \;\; + \frac{h^2}{2} f''(x_0) \;\; + \cdots$$

$$\frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2} f''(x_0) + \cdots$$

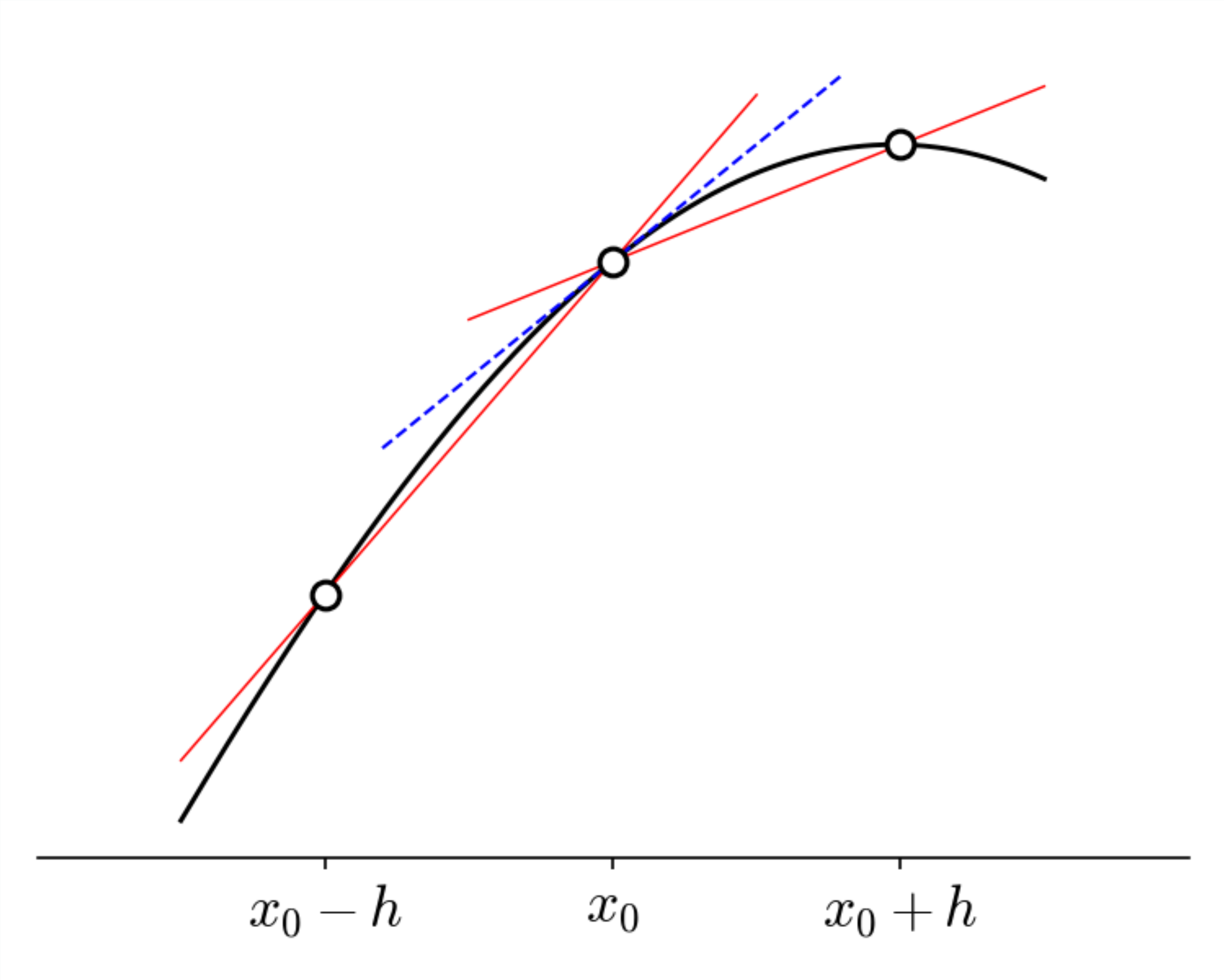
$$= f'(x_0) + \mathcal{O}(h)$$

truncate (ignore) higher order terms if h is small

truncation error

- Forward difference formula $$\left.\frac{\partial f}{\partial x}\right|_{x_0} \approx \frac{f(x_0 + h) - f(x_0)}{h}$$
- Backward difference formula $$\left.\frac{\partial f}{\partial x}\right|_{x_0} \approx \frac{f(x_0) - f(x_0 - h)}{h}$$

Numerical derivative



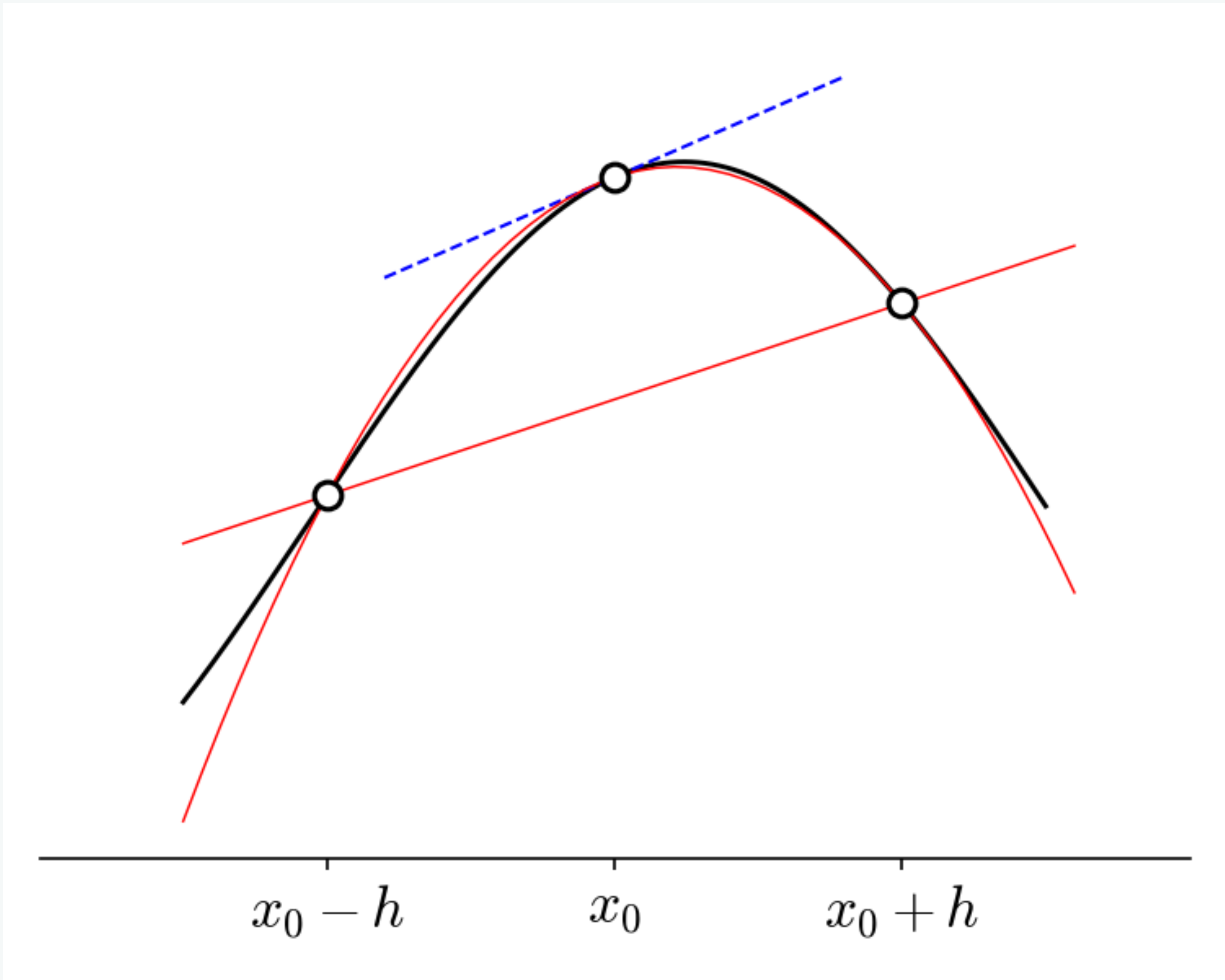
Error ~ $\mathcal{O}(h)$

first order accurate —
exact upto degree 1 polynomial (linear)

Numerical derivative

$$f(x_0 + h) = f(x_0) \;\; + hf'(x_0) \;\; + \frac{h^2}{2} f''(x_0) \;\; + \cdots$$

$$f(x_0 - h) = f(x_0) \;\; - hf'(x_0) \;\; + \frac{h^2}{2} f''(x_0) \;\; + \cdots$$



- Central difference (3-points) formula

$$\left.\frac{\partial f}{\partial x}\right|_{x_0} \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

Error ~ $\mathcal{O}(h^2)$ (second order accurate)

Numerical derivative

Higher-order accuracy formula

- 5 points

$$f' = \frac{f(x_0 - 2h) - 8f(x_0 - h) + 8(x_0 + h) - f(x_0 + 2h)}{12h} + \mathcal{O}(h^4)$$

- 7 points

$$f' = \frac{f(x_0 + 3h) - 9f(x_0 + 2h) + 45f(x_0 + h) - 45f(x_0 - h) + 9f(x_0 - 2h) - f(x_0 - 3h)}{60h} + \mathcal{O}(h^6)$$

note) more neighboring points ('stencil') are needed for higher order

Numerical derivative

Second derivatives

- 3-point

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \approx \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} + \mathcal{O}(h^2)$$

- 5-point

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \approx \frac{-f(x_0 - 2h) + 16f(x_0 - h) - 30f(x_0) + 16f(x_0 - h) - f(x_0 + 2h)}{12h^2} + \mathcal{O}(h^4)$$

Numerical derivative

Round-off error

Is it always better to have smaller h ?

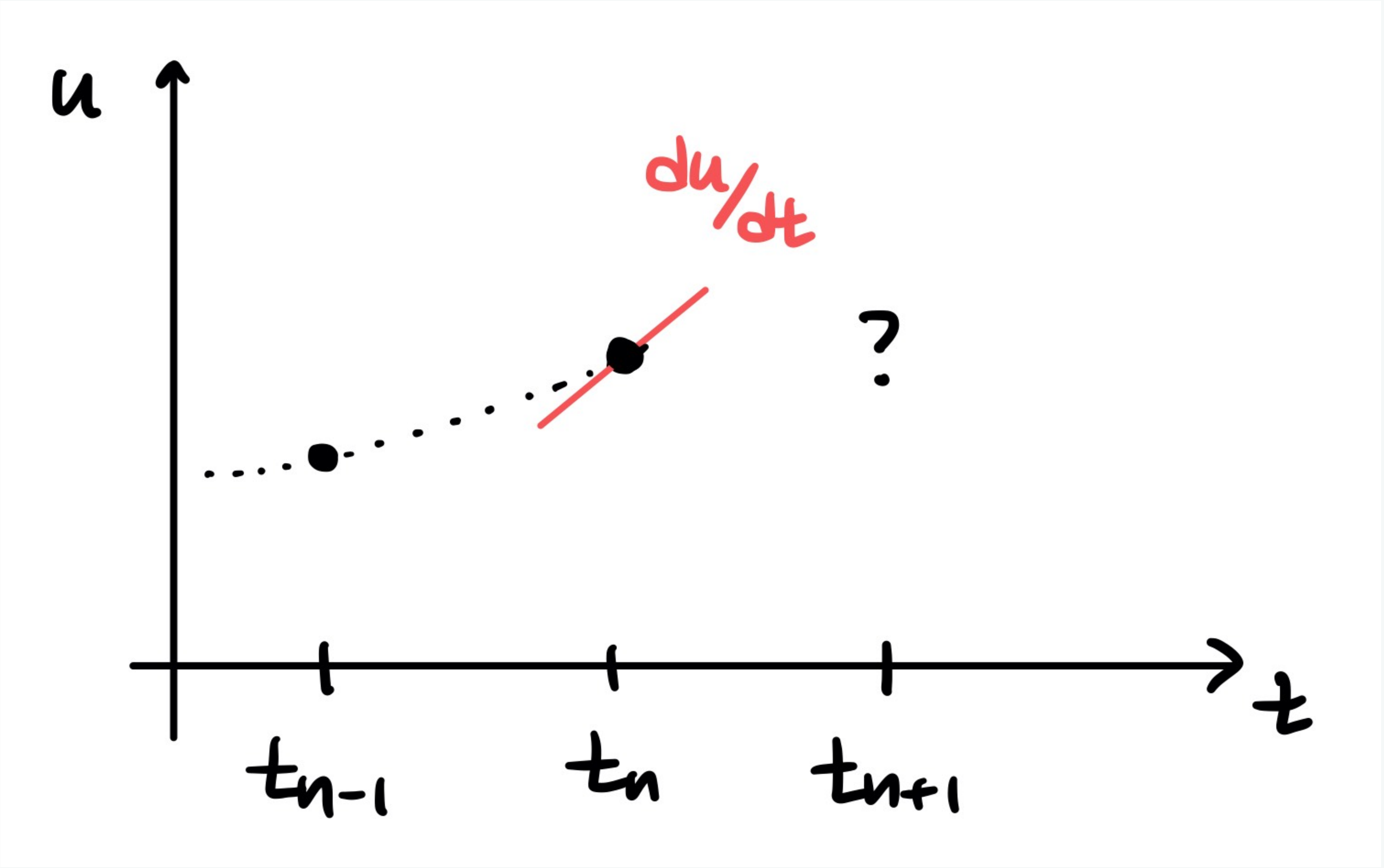
e.g. forward-difference numerical derivative of $f(x) = e^x$ at $x = 1$.

$$\frac{f(1+h) - f(1)}{h}$$

$$h = 10^{-k}, \quad k = 1,2,\cdots$$

```
for k =  1,   f'(1) = 2.858841954874,   error is  0.140560126415
for k =  2,   f'(1) = 2.731918655787,   error is  0.013636827328
for k =  3,   f'(1) = 2.719641422533,   error is  0.001359594074
for k =  4,   f'(1) = 2.718417747081,   error is  0.000135918622
for k =  5,   f'(1) = 2.718295419931,   error is  0.000013591472
for k =  6,   f'(1) = 2.718283187377,   error is  0.000001358918
for k =  7,   f'(1) = 2.718281965960,   error is  0.000000137501
for k =  8,   f'(1) = 2.718281825543,   error is -0.000000002916
for k =  9,   f'(1) = 2.718282054743,   error is  0.000000226284
for k = 10,   f'(1) = 2.718282053442,   error is  0.000000224983
```

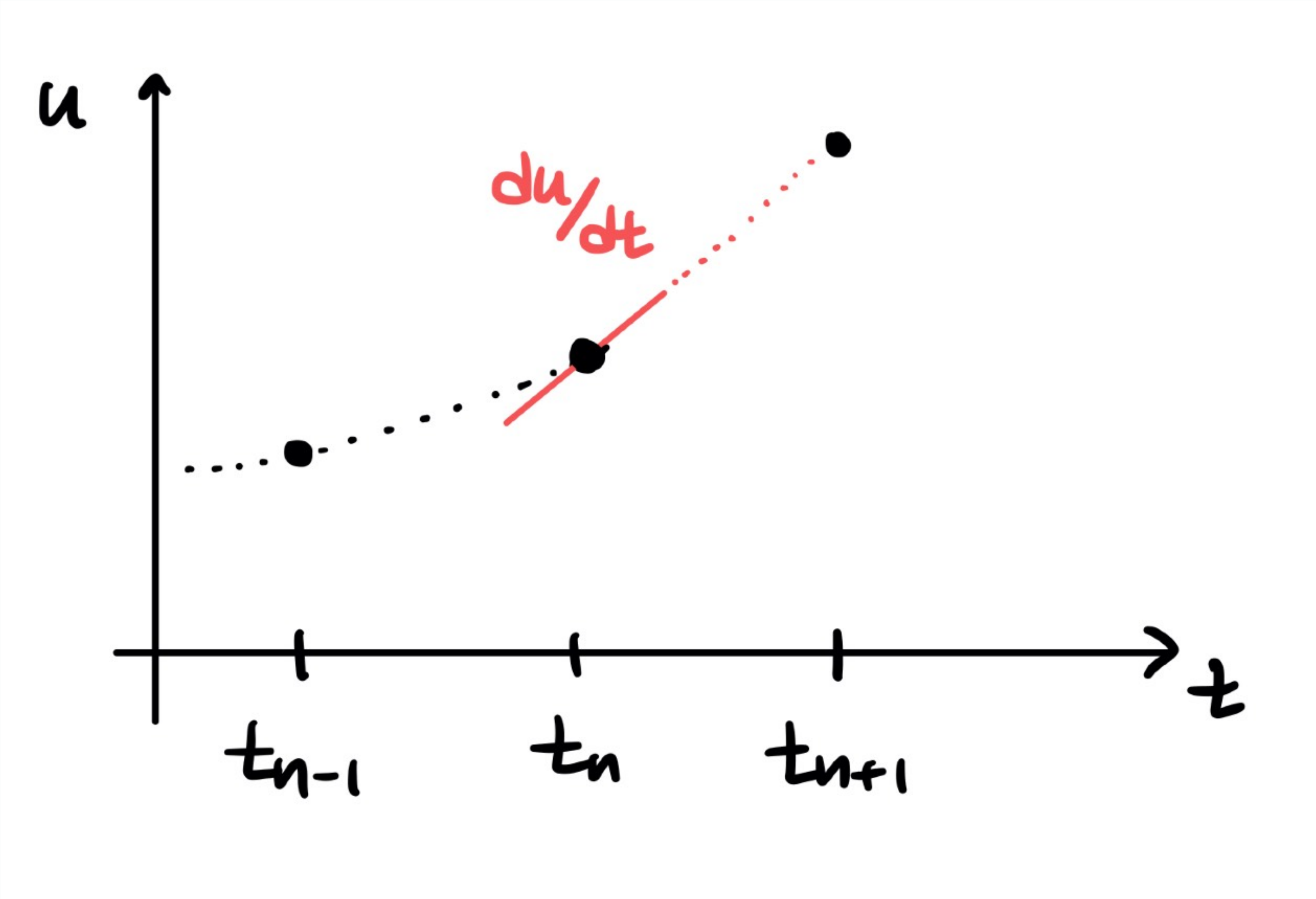
Time stepping



Time stepping

- Euler method (1st order)

$$u(x, t_{n+1}) \approx u(x, t_n) + \Delta t \left(\frac{\partial u}{\partial t} \right)_{(x,t_n)}$$

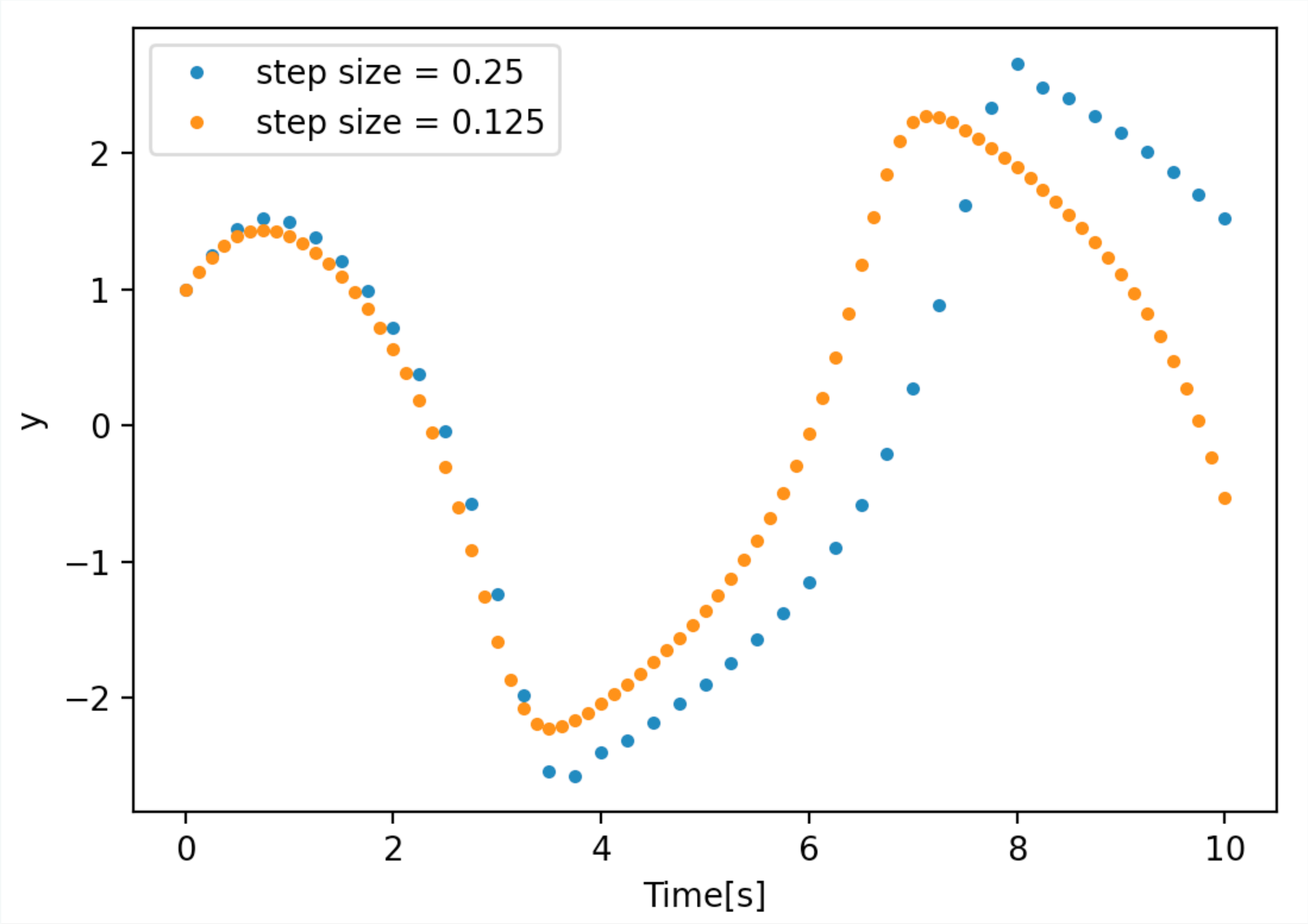


Time stepping

e.g. van der Pol equation

$$\frac{d^2y}{dt^2} - (1 - y^2)\frac{dy}{dt} + y = 0$$

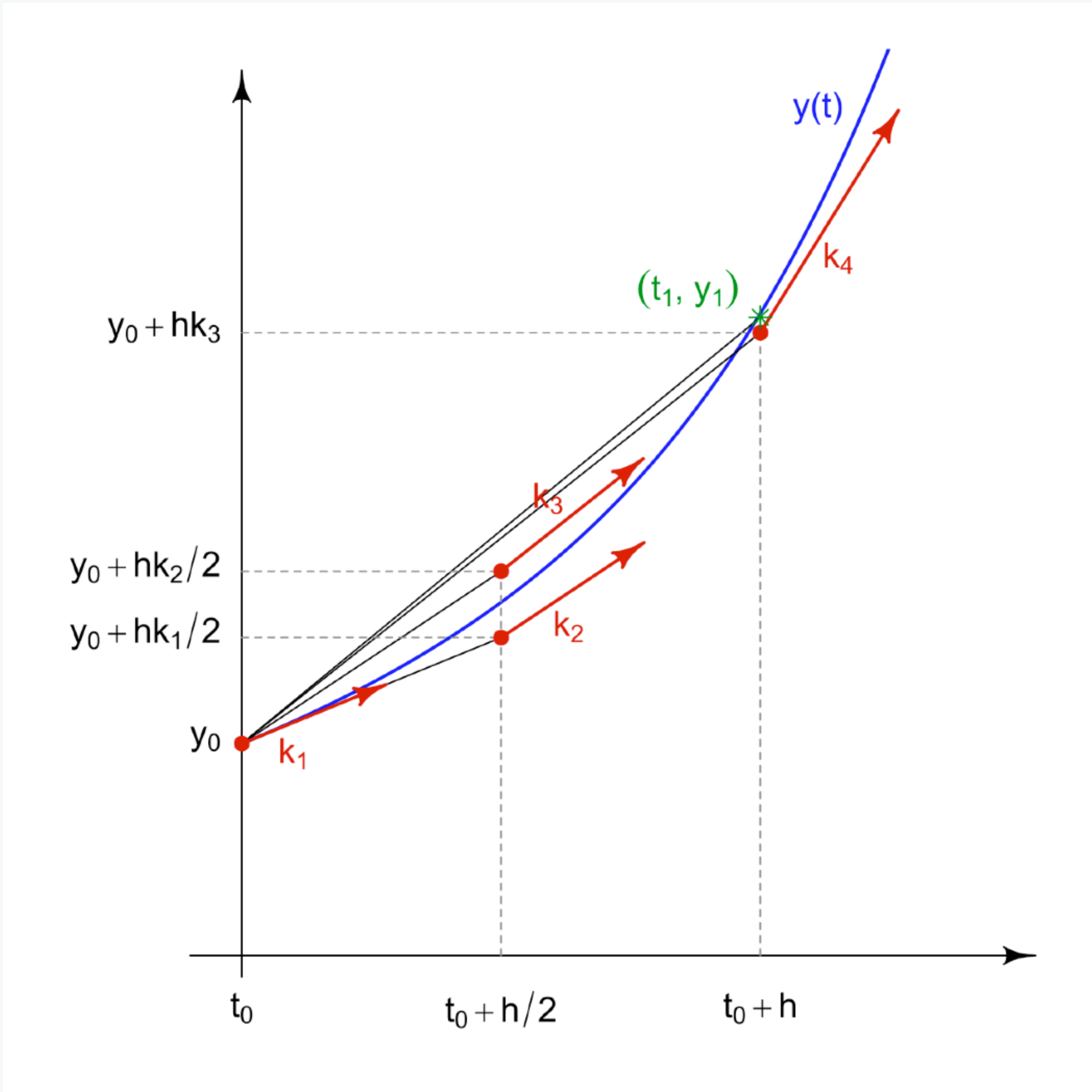
$$y(0) = y'(0) = 1$$



Time stepping

Higher order time stepping

e.g. Runge-Kutta methods

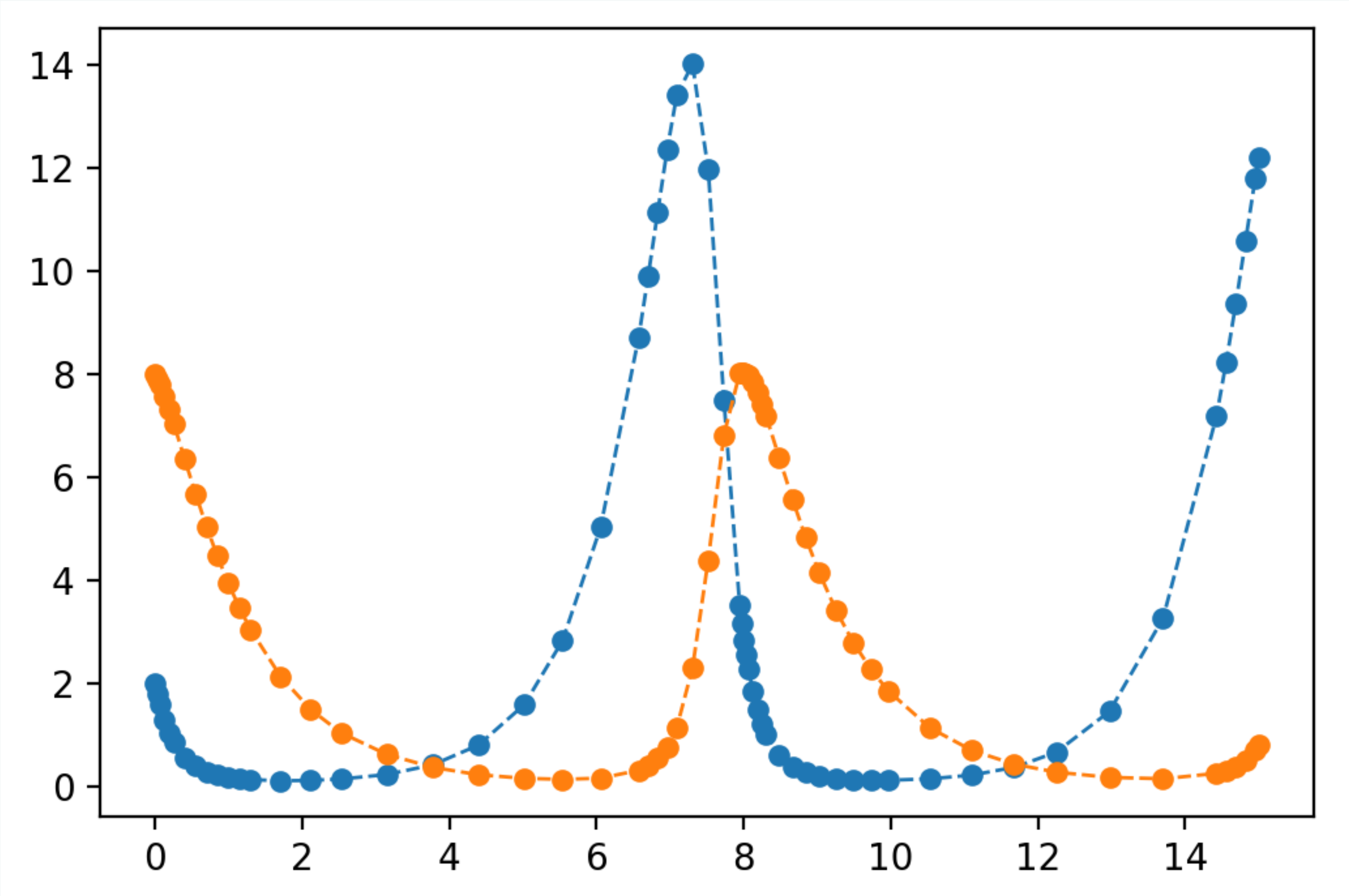


Credit : Wikipedia

Time stepping

Adaptive integrator

- changes step size with local truncation error



Time stepping

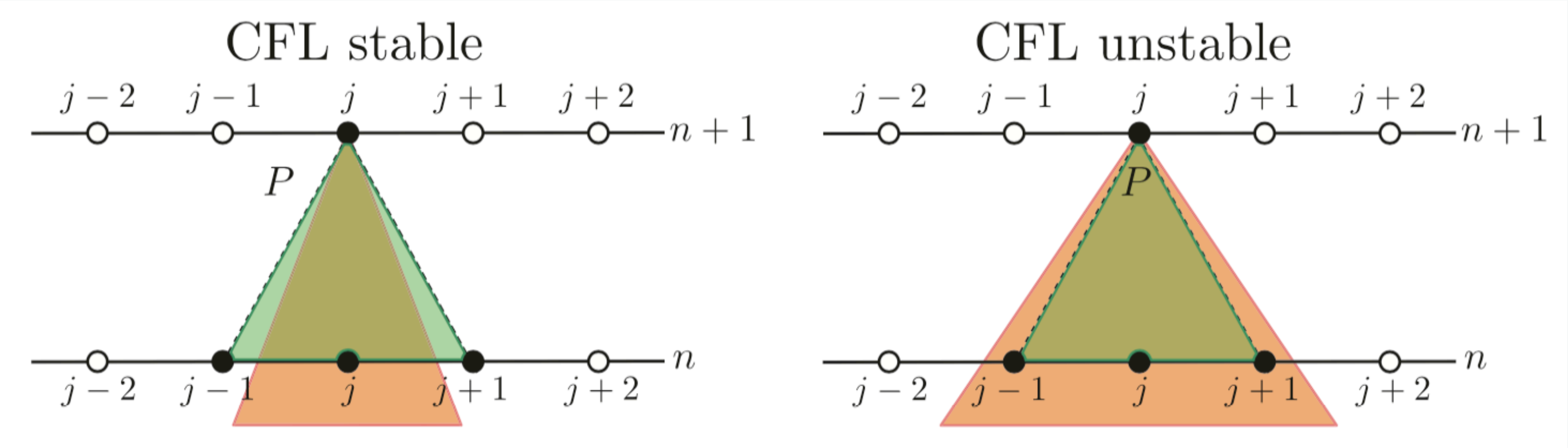
the Courant-Friedrichs-Lewy condition

- There is a max (limit) size of time step Δt

$$\Delta t_{\max} \sim \frac{C \Delta x}{\lambda}$$

CFL limit

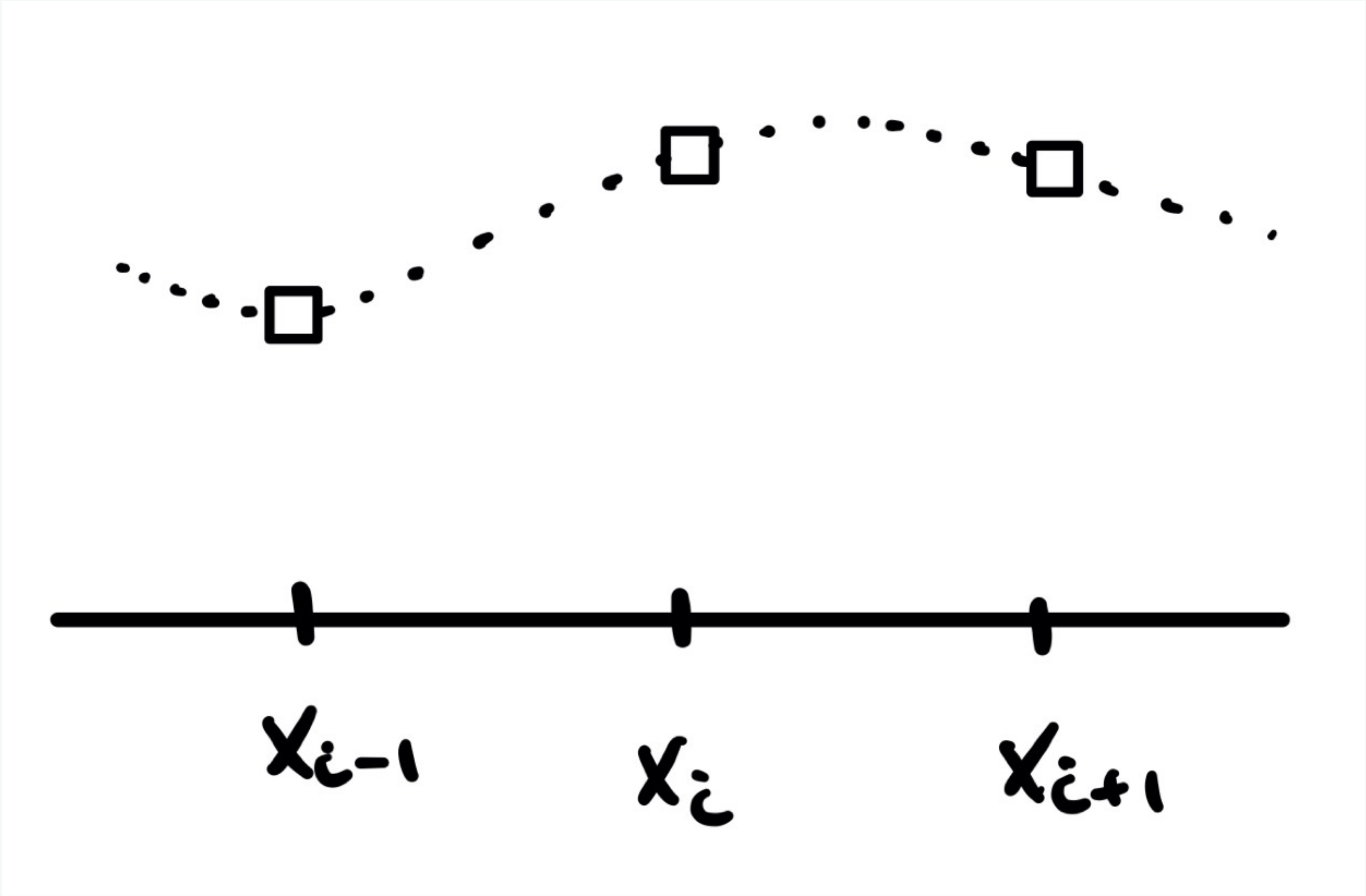
- *CFL number* : $\Delta t / \Delta t_{\max}$



Rezzolla & Zanotti 2013

Finite difference method

Numerical values $\{u_i\}$ represent pointwise values at $x = x_i$



- Fast & easy to implement
- Hard to use for complex geometries
  e.g. non-cartesian

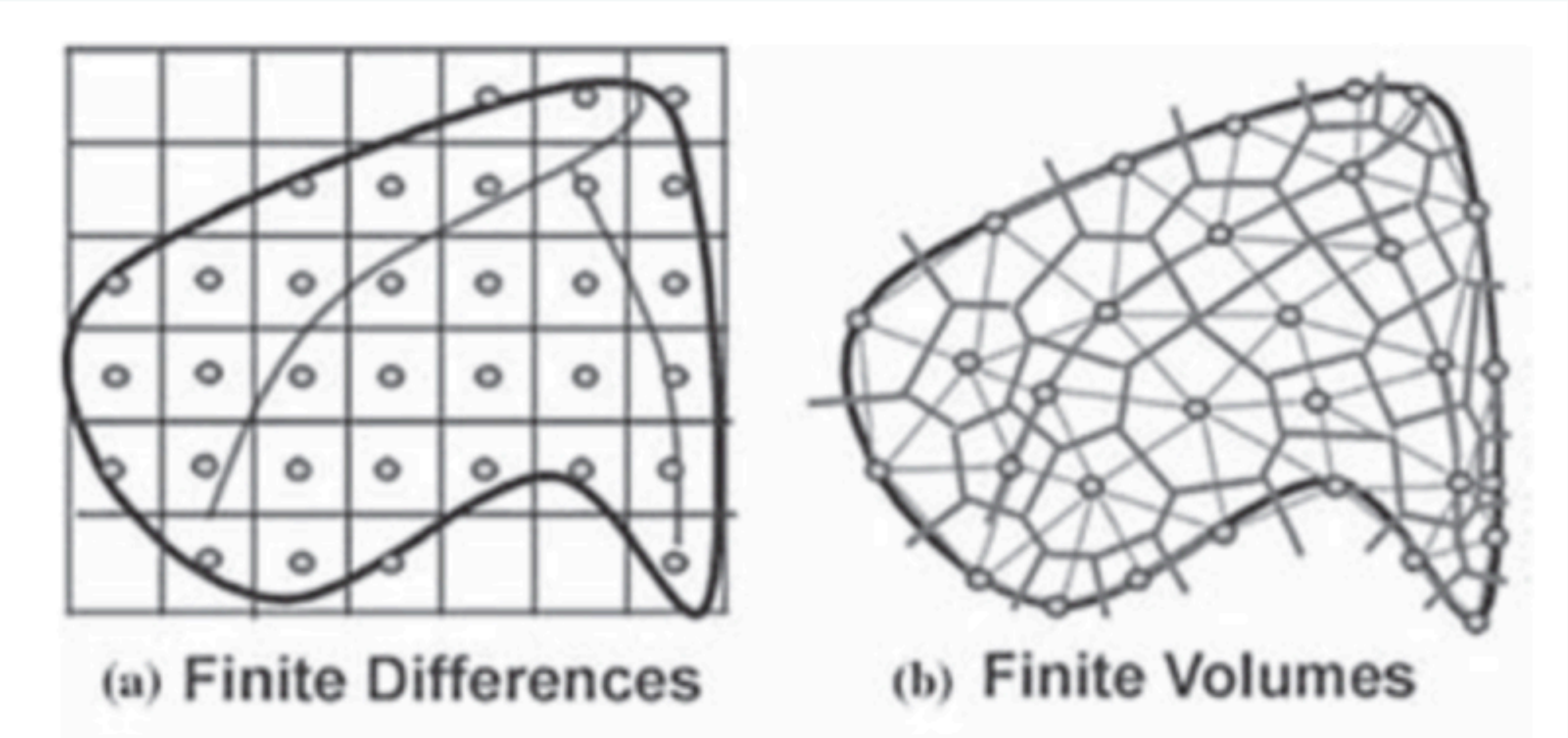
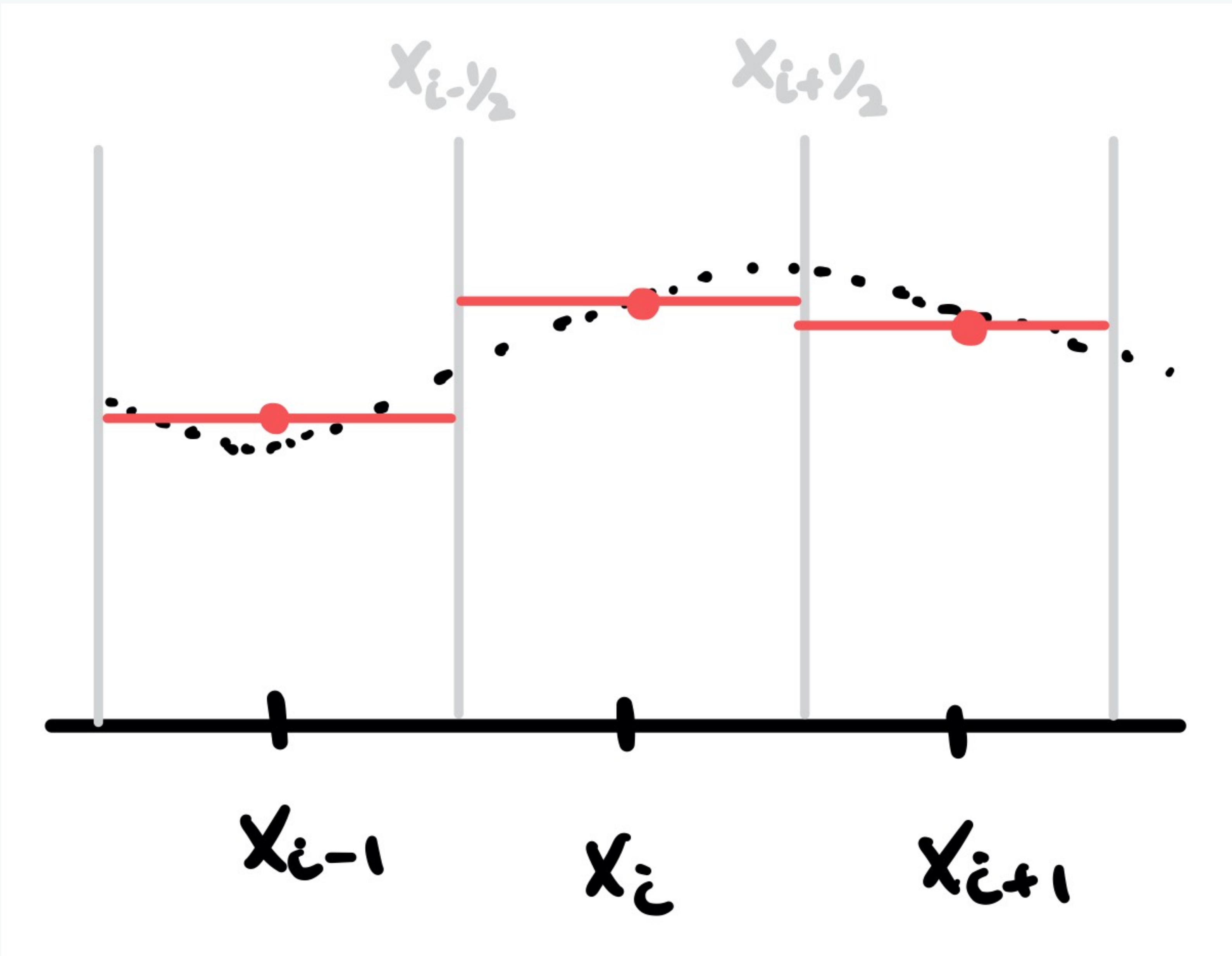


Image credit : https://ebrary.net/215164/environment/finite_differences

Finite volume method

Numerical values $\{u_i\}$ represent volume-averaged value over the cell $[x_{i-1/2}, x_{i+1/2}]$

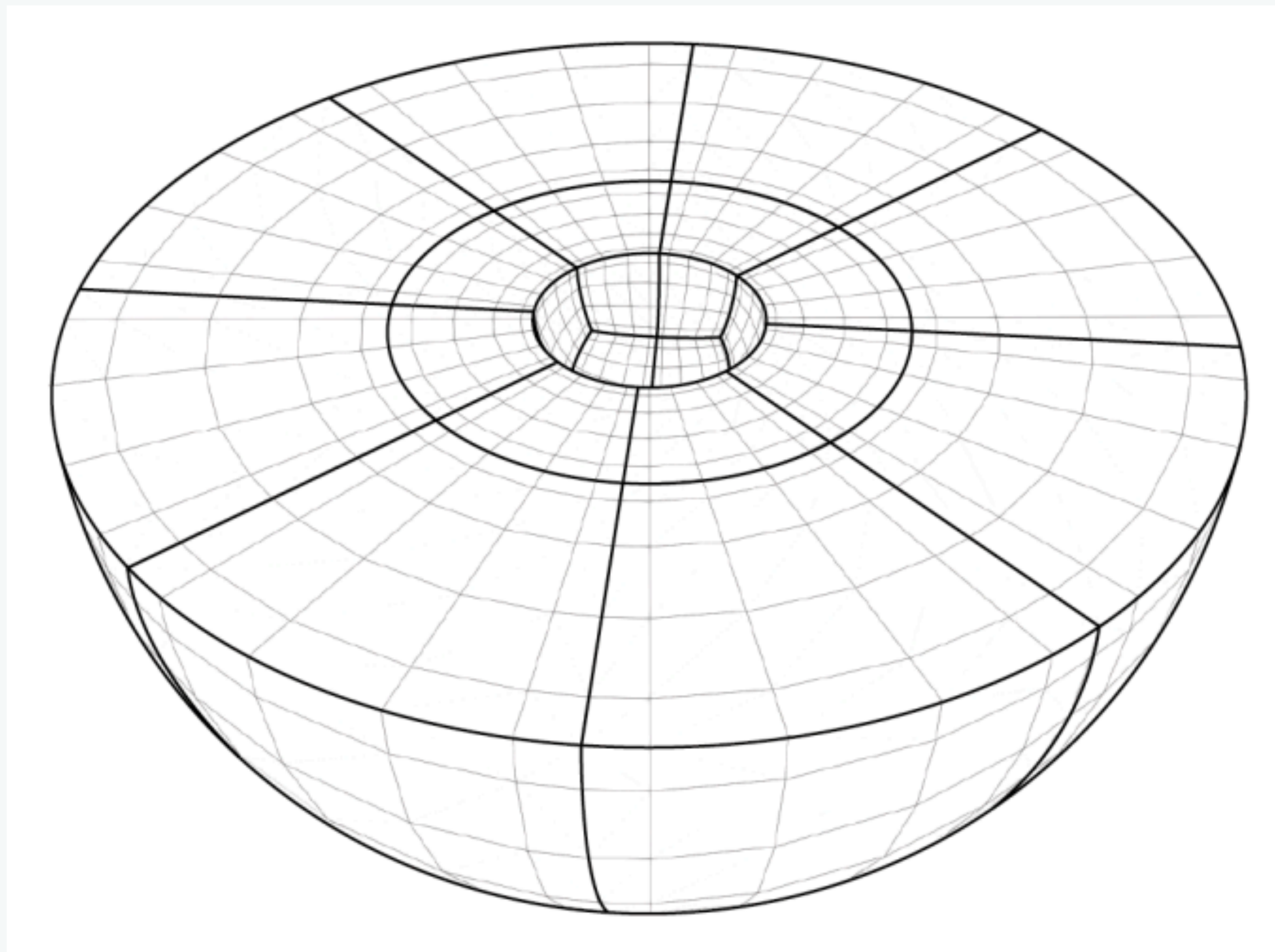


- **Useful for conservation-type equations, Resolves discontinuities very well**
  e.g. fluid dynamics
- **can handle unstructured mesh**
- **computationally more expensive**

Convergence of numerical methods

How many grid points do we need for desired accuracy?

- computational cost scales $\sim N^D$



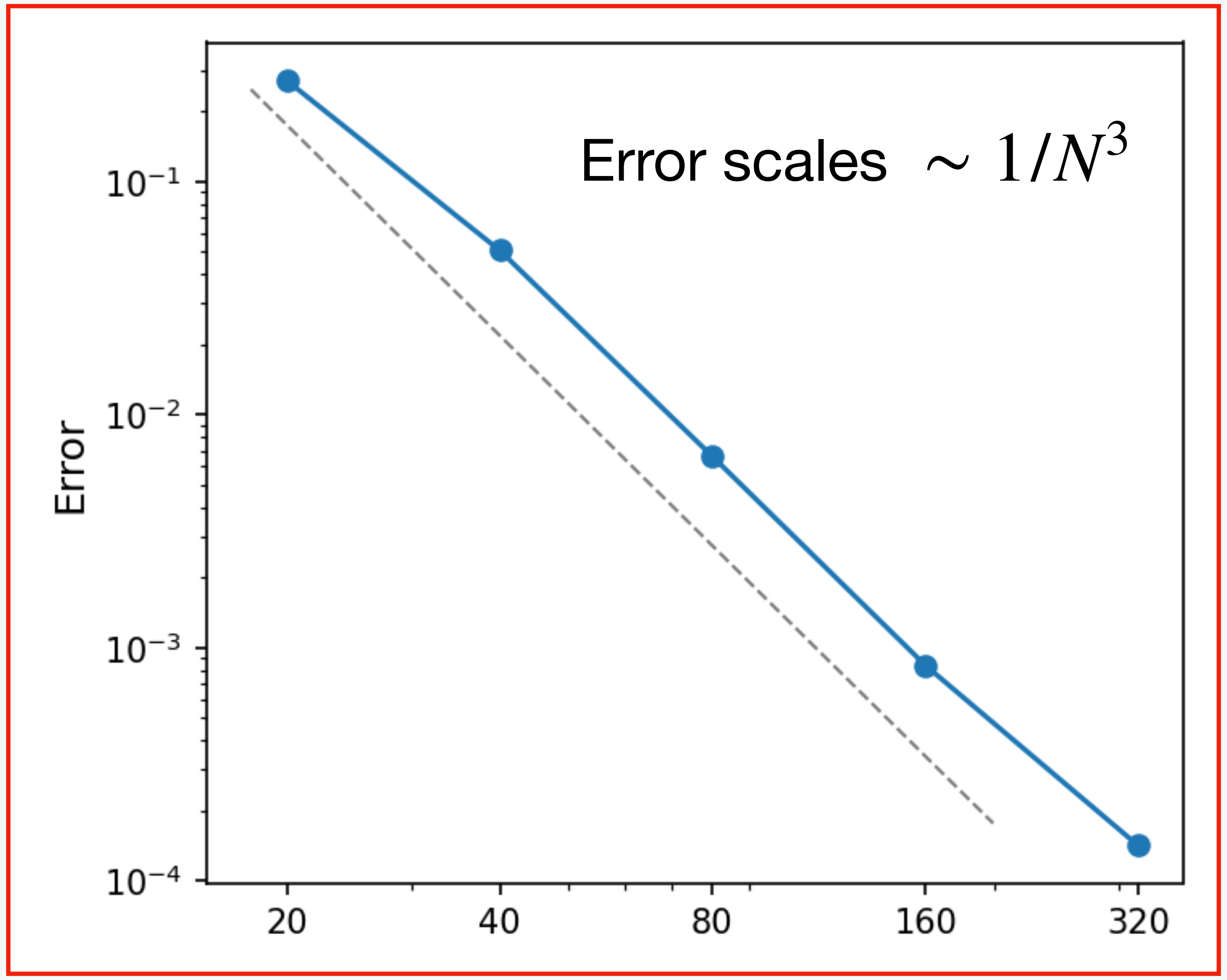
Vu+2022

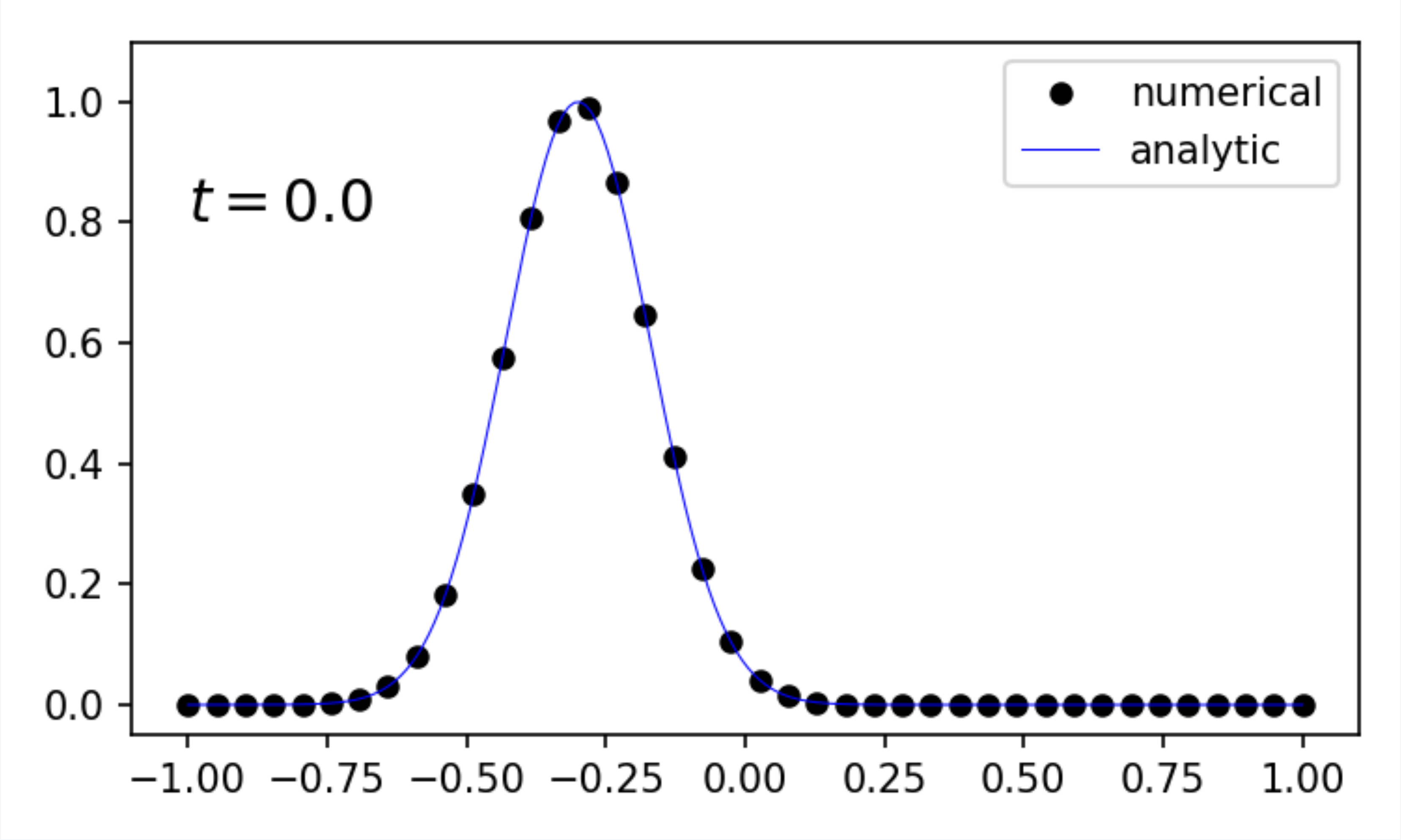
e.g. Advection equation $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$

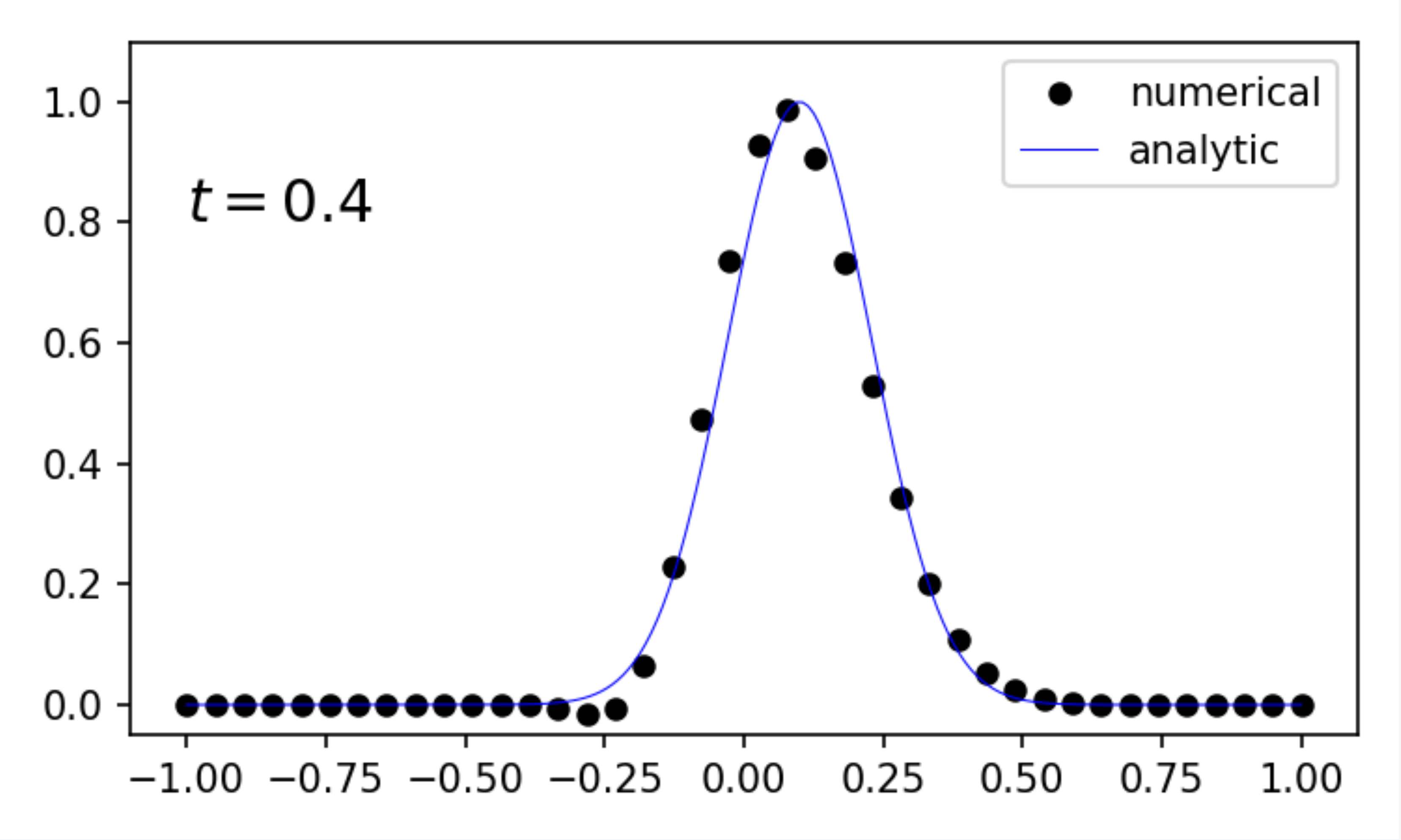
analytic solution : $u(x, t) = u_0(x - t)$

Solve numerically with:

$$u_i(t_{n+1}) = u_i - \left(\frac{u_{i+1} - u_{i-1}}{2\Delta x}\right)\Delta t$$







Convergence of numerical methods

n th-order accurate scheme : truncation error scales as h^{n+1} or $(N_{\rm grid})^{-(n+1)}$

Suppose we double (2x) the number of grid points per dimension, error drops…

- 1st order scheme : 1/4
- 2nd order scheme : 1/8
- …

* spectral (or exponential) accuracy : h^N or $\exp(-N_{\rm grid})$