1. FINITE-DIFFERENCE NUMERICAL METHODS FOR ODE SOLVING

- Start with something like: $\partial_t^2(u) = \mathrm{RHS}(u, t)$,   *this stands for "*<u>R</u>ight-<u>H</u>and <u>S</u>ide*"*
  - Re-formulate to coupled, first-order in time system $\partial_t u = v$, $\partial_t v = \mathrm{RHS}(u, t)$
- Discretize in time using finite time-steps: $f(t) \to f^n \equiv f(t_n), f^{n+1} \equiv f(t_n + \Delta t), \cdots$
- Starting from f^n, apply some update-rule to determine f^{n+1}
  - First-order explicit Euler step: $f^{n+1} = f^n + \Delta t \mathrm{RHS}(f^n, t_n)$
  - First-order *implicit* Euler step: $f^{n+1} = f^n + \Delta t \mathrm{RHS}(f^{n+1}, t_{n+1})$
  - Something better & higher-order with substeps (RK4, etc.)

2. GRID-BASED NUMERICAL METHODS FOR *PDE* SOLVING