Differential eqs.

- Ordinary diff. eqs. (ODEs) w/ boundary conditions
  ↳ Projects 1, 2
  - Finite diff. schemes → matrix problems
  - Shooting method

Now → ◉ ODEs w/ initial value conditions
  ↳ Project 3

- Partial diff. eqs. (PDEs) (multiple indep. variables)
  ↳ Project 5

ODEs w/ initial value conditions [Chap. 8]

- Recap classification:

- $$\frac{dy}{dt} = \underset{\substack{\uparrow \\ \text{known function}}}{f}(t, y)$$
  - First order: $\frac{dy}{dt}$
  - ODE: One indep. variable (t)
  - Init. value: $y(t_0)$

- $$\frac{d^2y}{dt^2} = f\left(t, \frac{dy}{dt}, y\right)$$
  - Second order: $\frac{d^2y}{dt^2}$
  - ODE: (t)
  - Init values: $y(t_0)$, $y'(t_0)$

Example: Newton's 2nd law:

$$\frac{d^2x}{dt^2} = \frac{1}{m} F\left(t, \frac{dx}{dt}, x\right)$$

- Second-order eqs. can often be rewritten as two first-order eqs.

Example

$$\frac{d^2x}{dt^2} = \frac{1}{m} F\left(t, \frac{dx}{dt}, x\right) \qquad (*)$$

- Define a new variable : $v \equiv \frac{dx}{dt}$ (conveniently named...)
- Can write $\frac{d^2x}{dt^2}$ as $\frac{dv}{dt}$
- Two eqs :

1) $\frac{dx}{dt} = v(t)$ from def.

2) $\frac{dv}{dt} = \frac{1}{m} F(t, v, x)$ from $(*)$

- Two coupled, first-order diff. eqs. for the variables x and v, both functions of a single indep. variable t.

- You should do this in Project 3

- Linear vs non-linear diff. eqs.:

$$\frac{dy}{dt} = g^3(t)\, y(t) \qquad \underline{\text{Linear}} \quad (y(t))$$

$$\frac{dy}{dt} = g^3(t)\, y(t) - h(t)\, y^2(t) \qquad \underline{\text{Non-linear}} \quad (y^2(t))$$

- In project 3, the Coulomb interaction between particles produce non-linear eqs.

Eq. for $\frac{d^2x}{dt^2}$ contains term $\propto \dfrac{x - x_j}{|\vec{r} - \vec{r}_j|^3} = \dfrac{x - x_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}^{\,3}}$

Non-lin. diff. eqs. typically requires a numerical approach! (An analytical sol. often does not exist.)

{ This type of info can be useful to mention in a report when motivating why you've studied something numerically }

Methods

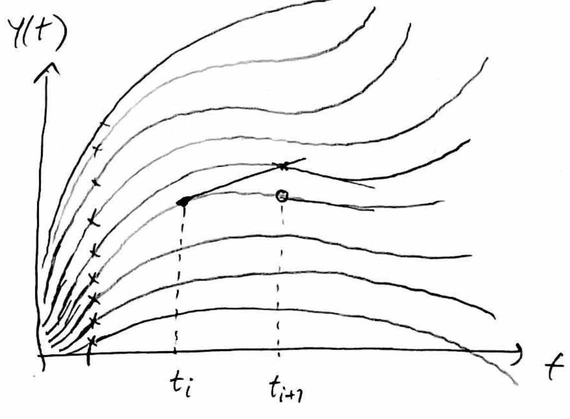
- Euler's forward method ✓
  - ↳ Euler-Cromer's method ✓
  - ↳ Midpoint-method
  - ↳ Half-step method
- Verlet and Leapfrog ✓
- Predictor-Corrector ✓
- Runge-Kutta ✓

- General considerations:
  - Local / global truncation errors?
  - Number of ~~e~~function evaluations needed in order to find next step y_{i+1}? ($\longleftrightarrow$ FLOPs) $f(t, y, y')$

  $\longrightarrow$ A balancing act! Some methods particularly useful for special problems

- There's typically an infinite number of solutions, and we basically jump between them when making approx.

$$\frac{dy}{dt} = f(t, y)$$



Initial conditions pick out one of an infinite number of solutions

Forward Euler

$$\boxed{\frac{dy}{dt} = f(t,y)}$$

- Look at a single first-order diff. eq.
- Init. value prob, we know $y(t_0)$
  want to find the corresponding solution $y(t)$

Algo

$$\boxed{y_{i+1} = y_i + hf_i}$$

- Taylor exp of $y(t+h)$, h small

$$y(t+h) = y(t) + y'(t)h + \mathcal{O}(h^2)$$

$$y(t+h) = y(t) + f(t,y)h + \mathcal{O}(h^2)$$

- Discretize

$$\Rightarrow y_{i+1} = y_i + f_i h + \mathcal{O}(h^2)$$

- Truncate

$$y_{i+1} \approx y_i + f_i h$$, truncation error $\mathcal{O}(h^2)$

- Algo. for approximating $y_1, y_2, y_3 \dots$, starting from y_0

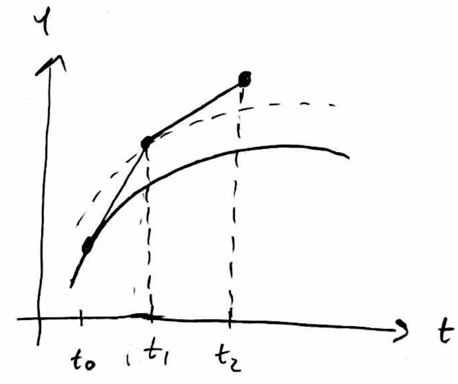
$$\boxed{y_{i+1} = y_i + hf_i}$$

- Local truncation error $\mathcal{O}(h^2)$ at each step
- A total of $n \propto \frac{1}{h}$ steps

$\Rightarrow$ Global error $\mathcal{O}(nh^2) = \mathcal{O}(h)$

+ very simple
+ basic element in many other algos.

- Single-step method. We only need the current point y_i to find the next.



$$\frac{d^2x}{dT^2} = f(t, x, \frac{dx}{dT})$$

Euler, coupled

- $v_{i+1} = v_i + h \cancel{\frac{dv}{dt}} f_i$
- $x_{i+1} = x_i + h v_i$

$v_i' = a_i = f_i$

$\dot{v} = f$

$\dot{x} = v$

Euler-Cromer

- $v_{i+1} = v_i + h f_i$
- $x_{i+1} = x_i + h v_{i+1}$

Predictor-Corrector method

- Simple improvement to Euler
- Also a single-step method (only requires knowing y_i)

Algorithm

1) Predict: $y^*_{i+1} = y_i + h f_i$

$\quad f^*_{i+1} = f(t_{i+1}, y^*_{i+1})$

2) Correct: $y_{i+1} = y_i + h \dfrac{f^*_{i+1} + f_i}{2}$

Alt. notation:

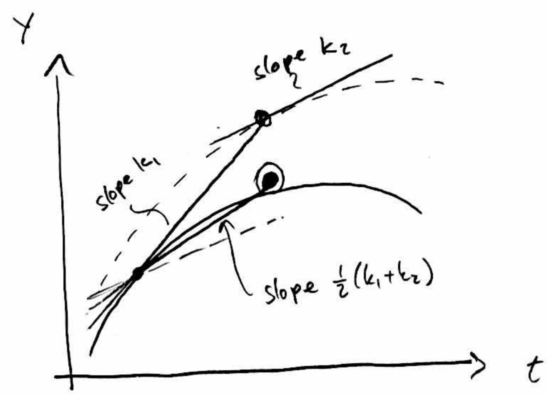
$k_1 \equiv h f_i = h f(t_i, y_i)$

$k_2 \equiv h f^*_{i+1} = h f(t_{i+1}, y^*_{i+1})$

$\Rightarrow \quad y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)$

- Euler uses gradient at a single point (f_i) to predict next point.
- Could improve by using <u>average</u> gradient between the two points t_i and t_{i+1}
- We <u>want</u> $y_{i+1} = y_i + h\left(\dfrac{f_i + f_{i+1}}{2}\right)$

but we don't know f_{i+1}. But we can <u>predict it</u> using a simple forward Euler step.



(Predictor - Corrector cont.)

- Local error (trunc.) is $\mathcal{O}(h^3)$ $\Rightarrow$ Global error $\mathcal{O}(h^2)$
- One order better than FE, but also req. one extra evaluation of f (at $f(t_{i+1}, y^*_{i+1})$)

Proof:

$$y(t+h) = y(t) + h y'(t) + \frac{1}{2} h^2 y''(t) + \underbrace{R_2(t)}_{\mathcal{O}(h^3)}$$

- Know from Mean Value Theorem that there exist a $\xi \in [t, t+h)$ such that

$$y(t+h) = \underset{\substack{\uparrow \\ \text{elegant!}}}{y(t)} + h y'(t) + \frac{1}{2} h^2 y''(t) + \frac{1}{3!} h^3 y'''(\xi)$$

- Use $y'(t) = f(t)$:

$$y(t+h) = y(t) + h f(t) + \frac{1}{2} h^2 f'(t) + \frac{1}{3!} h^3 f''(\xi)$$

- Replace $f'(t)$ with a forward diff. + remainder

$$f'(t) = \frac{f(t+h) - f(t)}{h} - \frac{1}{2} h f''(\eta)$$

$$\Rightarrow y(t+h) = y(t) + h f(t) + \frac{1}{2} h^2 \left[\frac{f(t+h) - f(t)}{h} - \frac{1}{2} h f''(\eta) \right] + \frac{1}{3!} h^3 f''(\xi)$$

$$= y(t) + h f(t) + \frac{1}{2} h \left[f(t+h) - f(t) \right] \underbrace{- \frac{1}{4} h^3 f''(\eta) + \frac{1}{3!} h^3 f''(\xi)}_{\mathcal{O}(h^3)}$$

$$\boxed{y(t+h) = y(t) + h \frac{f(t+h) - f(t)}{2} + \mathcal{O}(h^3)}$$

$$\boxed{y_{i+1} = y_i + h \frac{f_{i+1} - f_i}{2} + \mathcal{O}(h^3)}$$

Runge-Kutta

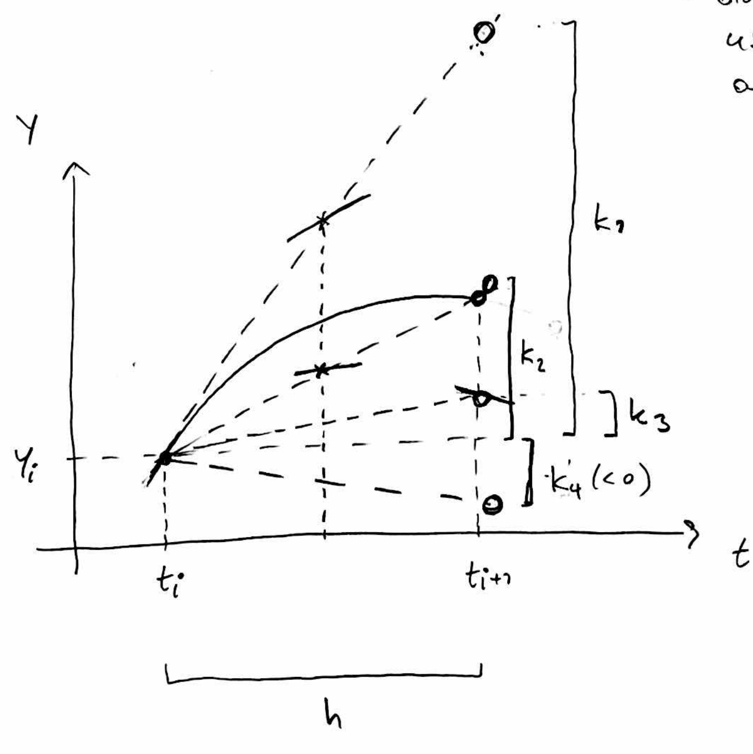
$$\frac{dy}{dt} = f(t, y)$$

- More "sophisticated" version of Predictor-Corrector
- An m-th order RK method uses m estimates of the gradient on the interval $t_i \le t \le t_{i+1}$ to determine Y_{i+1}
- Local error $O(h^{m+1}) \Rightarrow$ global error $O(h^m)$
- Classic choice: RK4

Algorithm:

1. $k_1 = h f(t_i, Y_i)$
2. $k_2 = h f\left(t_i + \frac{1}{2}h, Y_i + \frac{1}{2}k_1\right)$
3. $k_3 = h f\left(t_i + \frac{1}{2}h, Y_i + \frac{1}{2}k_2\right)$
4. $k_4 = h f\left(t_i + h, Y_i + k_3\right)$
5. $Y_{i+1} = Y_i + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right]$

- 4 evaluations of f
- Global error $O(h^4)$, so can use larger h than F.E. and P.C. $\Rightarrow$ More efficient



$$Y_{i+1} = Y_i + \frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4$$

RK4

$$y_{i+1} - y_i = \int_{t_i}^{t_{i+1}} \frac{dy}{dt}\, dt$$

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} \frac{dy}{dt}\, dt$$

$$\boxed{y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t,y)\, dt}$$

Solve with Simpson's rule

Simpson's rule

$$\int_a^b f(x)\, dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

$$= \frac{b-a}{6}\left[f(a) + 2f\left(\frac{a+b}{2}\right) + 2f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Two midpoint