

Linear ODEs, n^{th} order

Let $\mathcal{L}$ be an n^{th} order linear operator:

$$\mathcal{L} = \frac{d^n}{dx^n} + p_{n-1}(x) \frac{d^{n-1}}{dx^{n-1}} + \dots + p_0(x)$$

① The Homogeneous Equation

$$\mathcal{L}y = 0$$

has n linearly independent solutions.

$$y(x) = C^1 y_1(x) + C^2 y_2(x) + \dots + C^n y_n(x)$$

↑ basis of solutions ↑ coefs

Boundary conditions determine C_i .

② The Inhomogeneous Equation

$$\mathcal{L}y = f(x) \leftarrow \text{source}$$

has solutions of the form:

$$y(x) = \underbrace{y_p(x)}_{\substack{\text{particular} \\ \text{solution of} \\ \text{source}}} + \underbrace{C^1 y_1(x) + \dots + C^n y_n(x)}_{\text{solution of homogeneous eqn}}$$

Typical Strategy:

- ① Determine space of homogeneous solutions $y_1, \dots, y_n$
- ② Find/guess particular solution y_p
- ③ Fit boundary conditions to fix C^i

Linear ODEs w/ Constant Coefs

Any order. General form:

$$\textcircled{\star} \frac{d^ny}{dx^n} + a_{n-1} \frac{d^{n-1}y}{dx^{n-1}} + \dots + a_0 y = f(x) \quad \text{source}$$

Homogeneous eqn has exponential solutions: $y = e^{mx}$:

$$\frac{d^n}{dx^n} e^{mx} = m^n e^{mx}$$

Plugging in to $\textcircled{\star}$ divide out

$$(m^n + a_{n-1}m^{n-1} + \dots + a_0) \cancel{e^{mx}} = 0$$

$p(m)$ is n^{th} degree polynomial

Fundamental theorem of algebra $\Rightarrow$ n roots $\in \mathbb{C}$:

$$p(m) = (m - m_1)(m - m_2) \dots (m - m_n)$$

Each root gives lin ind solution:

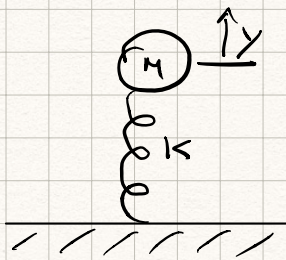
$$y(x) = C^1 e^{m_1 x} + C^2 e^{m_2 x} + \dots + C^n e^{m_n x}$$

If roots have multiplicity need to include derivatives of e^{mx} :

$$\frac{d}{dm} e^{mx} = x e^{mx}$$

to get full set of lin indep solutions.

Ex Hooke's Law



$$m \frac{d^2 y}{dt^2} = -ky \Rightarrow \left(\frac{d^2}{dt^2} + \frac{k}{m} \right) y = 0$$

homogeneous linear 2nd order constant coeffs

Plug in $e^{\alpha t}$;

$$\left(\alpha^2 + \frac{k}{m} \right) e^{\alpha t} = 0 \Rightarrow \alpha = \pm i \sqrt{\frac{k}{m}} = \pm i\omega$$

2 roots. General solution:

$$y(x) = C^1 e^{i\omega t} + C^2 e^{-i\omega t}$$

Or could use real basis:

$$\begin{aligned} y(x) &= C^1 (\cos \omega t + i \sin \omega t) + C^2 (\cos \omega t - i \sin \omega t) \\ &= \tilde{C}^1 \cos \omega t + \tilde{C}^2 \sin \omega t \end{aligned}$$

new constants

Boundary conditions determine specific solution:

Need 2 pieces of info to fix 2 constants

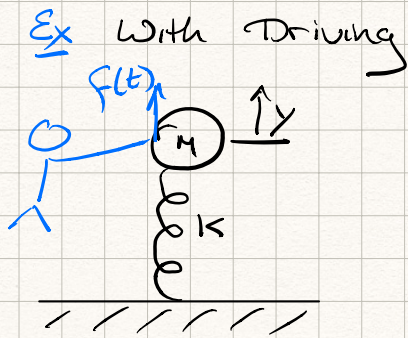
Initial position $\frac{1}{2}$ velocity

$$y(0) = 0 \quad \dot{y}(0) = v_0$$

$$\Rightarrow \tilde{C}_1 = 0 \quad \omega \tilde{C}_2 = v_0$$

$$y(t) = \frac{v_0}{\omega} \sin \omega t$$





$$m\ddot{y} = -ky + F(t)$$

Suppose $F(t) = F_0 \cos \Omega t$

$$\Rightarrow \left(\frac{d^2}{dt^2} + \frac{k}{m} \right) y = \frac{F_0}{m} \cos \Omega t$$

Guess particular solution $B \cos \Omega t$. Plug in:

$$B(-\Omega^2 + \frac{k}{m}) \cancel{\cos \Omega t} = \frac{F_0}{m} \cancel{\cos \Omega t}$$

$$B = \frac{F_0/m}{\omega^2 - \Omega^2}$$

So general solution

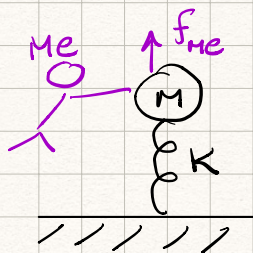
$$y(t) = \frac{F_0/m}{\omega^2 - \Omega^2} \cos \Omega t + C_1 \cos \omega t + C_2 \sin \omega t$$

Can fix C_1, C_2 by $y(0), \dot{y}(0)$.

N^{th} Order as N 1st Order

Can always rewrite N^{th} order ODE as N 1st order ODEs.

Ex Driven Spring



$$m \frac{d^2 y}{dt^2} = -ky + f_{me}(t)$$

2nd order, linear
inhomogeneous

Introduce: $p(t) = m \frac{dy}{dt}$

Equivalent to:

$$\begin{cases} \frac{dp}{dt} = -ky + f_{me}(t) \\ \frac{dy}{dt} = \frac{p}{m} \end{cases}$$

2 coupled 1st order

linear, inhomogeneous

→ Typically can introduce new dependent variable for each derivative to get N 1st order equations.

General form of N 1st Order ODEs

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \dots, y_N) \quad i = 1, \dots, N$$

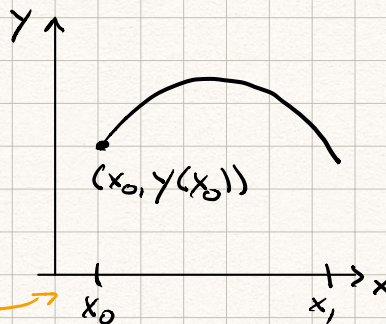
To specify solution, need boundary conditions:

- initial value problems

Specify $y_i(x_0)$ at some x_0

→ N parameters

Obtain $y_i(x)$ in region starting at x_0 by integrating forward



- boundary value problems

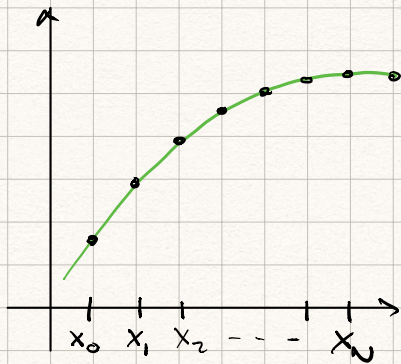
Specify y_i at 2 ends of region

(generally trickier)



Numerical Integration (Initial Value Problem)

Obtain approximate solution $y(x)$ on discrete grid of points x_n .



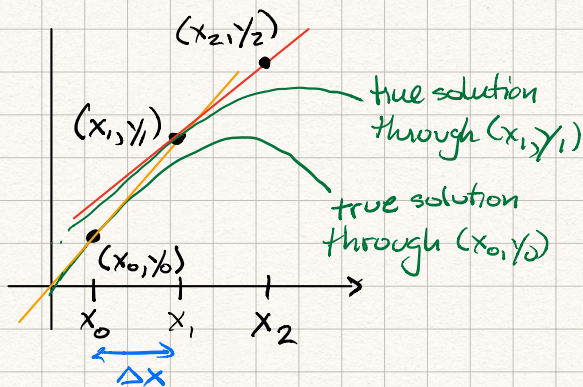
Step size Δx

- grid can be uniformly spaced but in practice better to adapt Δx to be finer where y varies faster

Many algorithms available...

Forward Euler Method (1st Order Runge-Kutta)

Conceptually important but neither accurate nor stable in practice



Approximate:

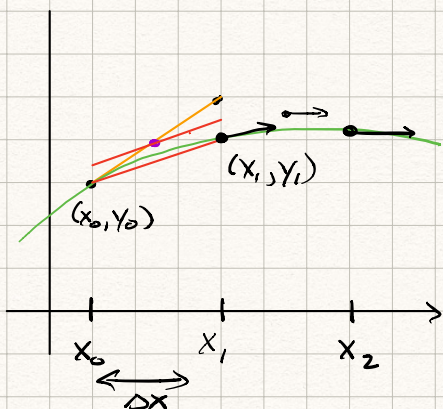
$$y(x_0 + \Delta x) = y(x_0) + \Delta x \underbrace{\frac{dy}{dx}}_{f(x_0, y(x_0))} \Big|_{x_0} + O(\Delta x^2)$$

So, given (x_n, y_n) , get next value:

$$y_{n+1} = y_n + \Delta x f(x_n, y_n)$$

- 1 evaluation of f per step
- Converges to ideal $y_n = y(x_n)$ as $\Delta x \rightarrow 0$
 - All methods better do this!
- Asymmetric: uses f at left edge of step only (forward)
- Error per step is $O(\Delta x^2)$, called a first order method

2nd Order Runge-Kutta



Use midpoint to make better estimate of y_{n+1} :

$$\Delta y_1 = \Delta x f(x_n, y_n)$$

$$\Delta y_2 = \Delta x f(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}\Delta y_1)$$

$$y_{n+1} = y_n + \Delta y_2 + O(\Delta x^3)$$

- 2 evaluations of f per step
- But: error is $O(\Delta x^3)$ (n^{th} order method has step error $O(\Delta x^{n+1})$)
[Can check by Taylor expansion]
- Can keep going to make more complicated series w/ higher orders

Work horse: 4th order Runge-Kutta

Additional features

- Adaptive step size is crucial:
 - take big steps where y is smooth & small where wiggly
 - Need to be able to detect errors.
- Stability: if steps too large, solutions may run away

Implementing a robust ODE integrator is tricky.

SciPy provides many good integrators. See HW.