Lecture 11: Ordinary Differential Equations (ODEs)

Initial Value Problem (IVP): Need to specify constraints at only one point (initial value) to produce unique solutions

Example: harmonic oscillator

Boundary Value Problem (BVP): Need to specify constraints at >1 points (on boundary) to produce unique solutions

Example: heat equation

Examples of Initial Value ODEs

$\frac{dx}{dt} = -ax$	1st order, linear
$\frac{dx}{dt} = ax(1-x)$	1st order, nonlinear
$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + ax = 0$	2nd order, linear
$\frac{d^2x}{dt^2} - \mu\left(1-x^2\right)\frac{dx}{dt} + x = 0$	2nd order, nonlinear

Solutions

$$\frac{dx}{dt} = -ax \quad \Rightarrow \quad x(t) = ce^{-ax}$$

$$\frac{dx}{dt} = ax(1-x) \quad \Rightarrow \quad x(t) = \frac{x_0}{(1-x_0)e^{-at} + x_0}$$

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + \omega_0^2 x = 0 \quad \Rightarrow \quad x(t) = ce^{-\beta t}\cos(\omega_1 t - \phi) \text{ for } \beta < \omega_0$$

$$\frac{d^2x}{dt^2} - \mu\left(1-x^2\right)\frac{dx}{dt} + x = 0 \quad \Rightarrow$$

no analytic solution -
use Matlab

Many n^{th} order ODEs can be written in terms of n 1^{st} order equations

Consider ODEs produced from F = ma:

$$F = ma \quad \Rightarrow \quad \frac{d^2x}{dt^2} = \frac{F(x,v,t)}{m} \quad \Rightarrow \quad \begin{cases} \dfrac{dx}{dt} = v \\ \dfrac{dv}{dt} = \dfrac{F(x,v,t)}{m} \end{cases}$$

Damped harmonic oscillator:

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + \omega_0^2 x = 0 \quad \Rightarrow \quad \begin{cases} \dfrac{dx}{dt} = v \\ \dfrac{dv}{dt} = -2\beta v - \omega_0^2 x \end{cases}$$

n^th order ODEs require n initial conditions (one for each 1^st order equation)

2nd order ODEs such as $F = ma$ require two initial conditions

$x(0)$ = initial position $\qquad \dfrac{dx}{dt} = v$

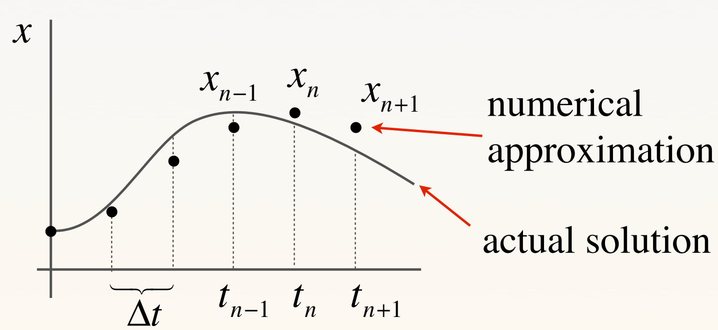
$v(0)$ = initial velocity $\qquad \dfrac{dv}{dt} = \dfrac{F(x,v,t)}{m}$

Notation

Break up time into discrete intervals Δt

Label the times as $t_n = n\Delta t$

Label the numerical solution as $x_n = x(t_n)$



Euler's Method

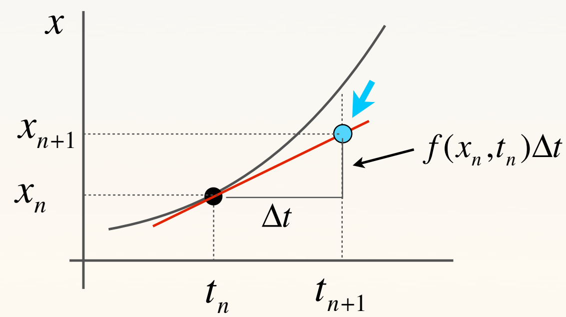
We want to solve a first-order equation of the form $\frac{dx}{dt} = f(x,t)$

Euler's method approximates the derivative as

$$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x(t_0 + \Delta t) - x(t_0)}{\Delta t} \qquad \text{or} \qquad \frac{dx}{dt} \approx \frac{x_{n+1} - x_n}{\Delta t}$$

Solve for x_{n+1}

$$\boxed{x_{n+1} = x_n + f(x_n, t_n)\Delta t}$$



Euler's Method

Write a program to solve $\frac{dx}{dt} = 2x$ for $0 \le x \le 3$

Pseudocode:

Initialization

Define: final time & time step Δt

Calculate number of points N

Preallocate arrays to store t and x values

set x(1) = initial value of x & t(1) = 0

Iteration

Create for loop to calculate t_n and x_n values

Present your results

Plot x vs. t

Euler's Method

Pros:

- easy to understand
- easy to implement

Cons:

- numerically unstable for lots of situations
- requires small time steps which introduces numerical round-off errors
- only 1st order accurate

Approximation Accuracy

Compare the numerical integration scheme to a Taylor series expansion:

$$x(t_n + \Delta t) = x(t_n) + \left.\frac{dx}{dt}\right|_{t_n} \Delta t + \frac{1}{2!}\left.\frac{d^2 x}{dt^2}\right|_{t_n} (\Delta t)^2 + \cdots$$

Euler $$\boxed{x(t_n + \Delta t) = x(t_n) + f(t_n)\Delta t} + \frac{1}{2!}\left.\frac{df(t)}{dt}\right|_{t_n} (\Delta t)^2 + \cdots$$

If the numerical integration method is equivalent to keeping n+1 terms in the Taylor series, then the integration scheme is said to be of n^{th} order.

Approximation Accuracy

Rule of Thumb:

If the local relative tolerance per time step is $\left|\frac{x_{numerical} - x_{actual}}{x_{actual}}\right| < 10^{-n}$

then it is best to choose at least an n^{th} order algorithm