Last Time:

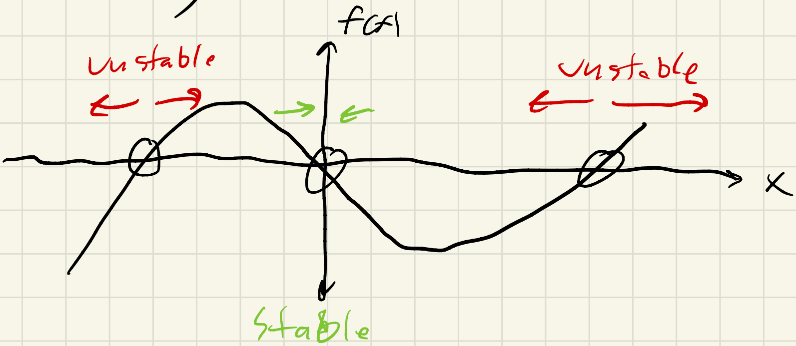
- Euler Integration
- Energy
- Lyapunov Stability
- Local Stability

Today:

- Finish Local Stability
- Taylor Integration
- Runge-Kutta Methods

---

* Local Stability



$$\frac{\partial f}{\partial x}\bigg|_{x^*} < 0 \Rightarrow \text{locally stable}$$

$$\frac{\partial f}{\partial x}\bigg|_{x^*} > 0 \Rightarrow \text{unstable}$$

$$\frac{\partial f}{\partial x}\bigg|_{x^*} = 0 \Rightarrow \text{inconclusive}$$

- This picture generalizes easily to $\mathbb{R}^n$

$\frac{\partial f}{\partial x} \in \mathbb{R}^{n \times n}$ is now a Jacobian matrix and we look at its Eigenvalues:

if **all** $\mathrm{Re}\left[\mathrm{eig}\left(\frac{\partial f}{\partial x}\Big|_{x^*}\right)\right] < 0 \Rightarrow$ locally stable

if **any** " $> 0 \Rightarrow$ unstable

if **any** " $= 0 \Rightarrow$ inconclusive

* For the Pendulum:

$$f(x) = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{\ell}\sin(\theta) \end{bmatrix} \Rightarrow \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell}\cos(\theta) & 0 \end{bmatrix}$$

- For the upward $(\theta = \pi)$ equilibrium we get:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ g/\ell & 0 \end{bmatrix} \Rightarrow \mathrm{eig}\left(\frac{\partial f}{\partial x}\right) = \pm\sqrt{\frac{g}{\ell}} \Rightarrow \text{unstable}$$

- For the downward $(\theta = 0)$ equilibrium:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -g/\ell & 0 \end{bmatrix} \Rightarrow \mathrm{eig}\left(\frac{\partial f}{\partial x}\right) = 0 \pm i\sqrt{\frac{g}{\ell}} \Rightarrow \text{inconclusive}$$

(but we know already its Lyapunov stable)

* Local Stability in Discrete Time:

- For a discrete-time system

$$x_{n+1} = f(x_n)$$

- Equilibrium $\Rightarrow$ $x^* = f(x^*)$ (fixed point)

- Think about an iterated map:

$$x_N = f(f(f(f(\cdots f(x_0))))))$$

- Stable $\Rightarrow$ $\lim_{N\to\infty} x_N = f(f(f(\cdots f(x^* + \underbrace{\varepsilon}_{\text{"small"}})))) = x^*$

- Assuming small ε, use the chain rule:

$$f(x^* + \varepsilon) \approx f(x^*) + \left.\frac{\partial f}{\partial x}\right|_{x^*} \varepsilon$$

$$\Rightarrow x_N \approx x^* + \left(\left.\frac{\partial f}{\partial x}\right|_{x^*}\right)^{N-1} \underbrace{\varepsilon}_{\text{arbitrary}} = x^* \;\; \forall \varepsilon$$

$\Rightarrow$ if <u>all</u> $\left|\operatorname{eig}\left(\left.\frac{\partial f}{\partial f}\right|_{x^*}\right)\right| < 1 \Rightarrow$ locally stable

if <u>any</u> " $> 1 \Rightarrow$ unstable

if <u>any</u> " $= 1 \Rightarrow$ inconclusive

* Higher-Order Integrators

- Explicit Euler is equivalent to taking 1st order Taylor expansion of $X(t_{n+1})$ about $X(t_n)$:

$$X_{n+1} = X(t_n + \underbrace{h}_{\text{time step}}) \approx X_n + \underbrace{\frac{dx}{dt}\Big|_{x_n} h}_{hf(x_n)} + \ldots$$

- An obvious way to get a better answer is to take more terms in this expansion

$$X_{n+1} \approx X(t_n) + h\frac{dx}{dt}\Big|_{x_n} + \frac{h^2}{2}\frac{dx^2}{dt^2}\Big|_{x_n} + \frac{h^3}{6}\frac{d^3x}{dt^3} + \ldots$$

- Expanding the derivatives in terms of $f(x)$:

$$\frac{dx}{dt} = f(x)$$

$$\frac{d^2x}{dt^2} = \frac{d}{dt}f(x) = \frac{\partial f}{\partial x}\frac{dx}{dt} = \frac{\partial f}{\partial x}f(x)$$

$$\frac{d^3x}{dt^2} = \frac{d}{dt}\left[\frac{\partial f}{\partial x}f(x)\right] = \frac{\partial f}{\partial x}\frac{\partial f}{\partial x}f(x) + \left[\underbrace{\frac{\partial^2 f}{\partial x^2}}_{\text{3rd rank Tensor}}f(x)\right]f(x)$$

- In theory, as long as $f(x)$ is smooth, we can compute solutions to arbitrary accuracy this way.

- Not very popular due to need to compute higher-order derivatives of $f(x)$ (high-rank tensors)

- Definition: "order of accuracy" of an ODE solver is the number of Taylor series terms that its approximate solution matches.

$\Rightarrow$ Euler is a 1^{st}-order method

* Runge-Kutta Methods:

- We want to achieve high order without computing higher derivatives of $f(x)$
- Key idea: use multiple evaluations of $f(x)$ over the time step to fit a polynomial to $x(t)$
- Number of $f(x)$ evaluations per time step = number of "stages"
- What's the best we can do with one stage?

$$x_{n+1} \approx x_n + hf((1-\alpha)x_n + \alpha x_{n+1}), \quad 0 \le \alpha \le 1$$

$$\approx x_n + hf(x_n) + \alpha h^2 \frac{\partial f}{\partial x} f(x) + O(h^3)$$

$\Rightarrow \alpha = 1/2$ gives a 2^{nd}-order method!

* Implicit Midpoint:

$$x_{n+1} = x_n + hf(\tfrac{1}{2}x_n + \tfrac{1}{2}x_{n+1})$$

- x_{n+1} appears on RHS inside $f(x)$ $\Rightarrow$ "implicit"

- We need to solve for x_{n+1} with Newton's method:

$$r(x_{n+1}) = x_{n+1} - x_n - h f(1/2 x_n + 1/2 x_{n+1}) = 0$$

$$r(x_{n+1} + \Delta x_{n+1}) \approx r(x_{n+1}) + \frac{\partial r}{\partial x}\Big|_{x_{n+1}} \Delta x_{n+1} = 0$$

$$\Rightarrow \Delta x_{n+1} = -\left(\frac{\partial r}{\partial x}\Big|_{x_{n+1}}\right)^{-1} r(x_{n+1})$$

$$x_{n+1} \leftarrow x_{n+1} + \Delta x_{n+1}$$

(repeat until convergence)

(can use $x_{n+1} \approx x_n$ as initial guess)

- Generally converges in 3~4 iterations
- Downside: solving root-finding problem is expensive

* Explicit Midpoint
  - Avoid root-finding
  - Let's evaluate the midpoint using an Euler step:

$$x_{n+1} = x_n + h f(\underbrace{x_n + h/2 f(x_n)}_{\text{approximate midpoint}})$$

- Taylor expand:

$$x_{n+1} \approx x_n + hf(x_n) + \frac{h^2}{2}\frac{\partial f}{\partial x} f(x) + O(h^3)$$

$\Rightarrow$ Explicit midpoint is 2^{nd} order

- Explicit midpoint has 2 stages