Last Time:

- Stability
- Discrete-time simulation
- Forward / backward Euler
- RK4

Today:

- Notation
- Root Finding
- Minimization

Some Notation:

- Given $f(x) : \mathbb{R}^n \to \mathbb{R}$

$$\frac{\partial f}{\partial x} \in \mathbb{R}^{1 \times n} \quad \text{is a } \underline{\underline{\text{row vector}}}$$

- This is because $\frac{\partial f}{\partial x}$ is the linear operator mapping Δx into Δf:

$$f(x+\Delta x) \approx f(x) + \frac{\partial f}{\partial x} \Delta x$$

- Similarly $g(y) : \mathbb{R}^m \to \mathbb{R}^n$

$$\frac{\partial g}{\partial y} \in \mathbb{R}^{n \times m} \quad \text{because:}$$

$$g(y+\Delta y) \approx g(y) + \frac{\partial g}{\partial y} \Delta y$$

- These conventions make the chain rule work:

$$f(g(y+\Delta y)) \approx f(g(y)) + \frac{\partial f}{\partial x}\Big|_{g(y)} \frac{\partial g}{\partial y}\Big|_{y} \Delta y$$

- For convenience, we will define:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x}\right)^T \in \mathbb{R}^{n \times 1} \quad \underline{\text{column vector}}$$

$$\nabla^2 f(x) = \frac{\partial}{\partial x}(\nabla f(x)) = \frac{\partial^2 f}{\partial x^2} \in \mathbb{R}^{n \times n}$$

$$f(x+\Delta x) \approx f(x) + \frac{\partial f}{\partial x}\Delta x + \frac{1}{2}\Delta x^T \frac{\partial^2 f}{\partial x^2}\Delta x$$

Root Finding:

- Given $f(x)$, find x^* such that $f(x^*) = 0$

* Example: equilibrium of a continuous-time dynamics
  - Closely related: fixed point:

$$f(x^*) = x^*$$

(equilibrium of discrete-time dynamics)

* Fixed-Point Iteration
  - Simplest solution method
  - If fixed point is stable, just "iterate the dynamics" until it converges
  - Only works if x^* is a stable equilibrium point and if initial guess is in the basin of attraction
  - Can converge slowly (depends on f)

* Newton's Method:

- Fit a linear approximation to $f(x)$:

$$f(x+\Delta x) \simeq f(x) + \frac{\partial f}{\partial x}\bigg|_x \Delta x$$

- Set approximation to zero and solve for Δx:

$$f(x) + \frac{\partial f}{\partial x}\Delta x = 0 \Rightarrow \Delta x = -\left(\frac{\partial f}{\partial x}\right)^{-1} f(x)$$

- Apply correction:

$$x \leftarrow x + \Delta x$$

- Repeate until Convergence

* Example: Backward Euler

$$f(x_{n+1}, x_n, u_n) = 0$$

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

evaluate f at future time

$$\Rightarrow f(x_{n+1}, x_n, u_n) = x_{n+1} - x_n - h f(x_{n+1}) = 0$$

- Very fast convergenee with Newton (Quatratic)
- Can get machine precision
- Most expensive part is solving a linear system $O(n^3)$
- Can improve complexity by taking advantage of problem structure/sparsity (more later)

Minimization:

$$\min_x f(x) \quad , \quad f(x): \mathbb{R}^n \to \mathbb{R}$$

- If f is smooth, $\frac{\partial f}{\partial x}\Big|_{x^*} = 0$ at a local minimum
- Now we have a root-finding problem $\nabla f(x) = 0$

$\Rightarrow$ Apply Newton!

$$\nabla f(x + \Delta x) \approx \nabla f(x) + \underbrace{\frac{\partial}{\partial x}(\nabla f(x))}_{\nabla^2 f} \Delta x = 0$$

$$\Rightarrow \Delta x = -(\nabla^2 f(x))^{-1} \nabla f(x)$$

$$x \leftarrow x + \Delta x$$

repeate until convergence

* Intuition:
  - Fit a quadratic approximation to $f(x)$
  - Exactly minimize approximation

* Example:

$$\min_x f(x) = x^4 + x^3 - x^2 - x$$

- Start at: $1.0, -1.5, 0$ } maximizes!

★ Take-away Messages:

- Newton is a <u>local</u> <u>root-finding</u> method. Will converge to the closest fixed point to the initial guess (min, max, saddle)

★ Sufficient Conditions

- $\nabla f = 0$ "first-order necessary condition" for a minimum. Not a sufficient condition.

- Let's look at scalar case:

$$\Delta x = \underset{\text{"descent"}}{-}\underbrace{(\nabla^2 f)^{-1}}_{\text{"learning rate"/"step size"}}\underbrace{\nabla f}_{\text{gradient}}$$

$$\nabla^2 f > 0 \Rightarrow \text{descent (minimization)}$$

$$\nabla^2 < 0 \Rightarrow \text{ascent (maximization)}$$

- In $\mathbb{R}^n$, $\nabla^2 f > 0$, $\nabla^2 f \in S^n_{++}$

(positive definite) $\Rightarrow$ descent

- If $\nabla^2 f > 0$ everywhere $(\forall x) \Leftrightarrow f(x)$ is strongly convex

$\Rightarrow$ Can always solve with Newton

- Usually not the case for hard/nonlinear problems

* Regularization:

  - Practical solution to make sure we always minimize:

$$H \leftarrow \nabla^2 f$$

$$\text{while } H \not\succ 0 \quad \text{"not pos. def."}$$

$$\quad H \leftarrow H + \beta I \qquad \text{scalar hyperparameter } (\beta > 0)$$

$$\text{end}$$

$$\Delta x = -H^{-1} \nabla f$$

$$x \leftarrow x + \Delta x$$

- Also called "damped Newton" (shrinks steps)
- Guarantees descent

* Example:

  - Regularization makes sure we minimize
  - What about overshoot?