There are two common ways for a numerical computation to go wrong.

**CONDITIONING**

The problem is sensitive to small changes.

**STABILITY**

The algorithm introduces large errors.

---

**Definition**

Let \( y = f(x) \). The relative condition number of \( f \) tells us how much \( y \) changes if we change \( x \) by a tiny bit. Let \( \hat{y} = f(\hat{x}) \), where \( \hat{x} \approx x \). Then the relative condition number is

\[
\kappa(x) = \left| \frac{xf'(x)}{f(x)} \right|.
\]

Why is this right? We want

\[
\left| \frac{\hat{y} - y}{y} \right| \approx \kappa(x) \left| \frac{\hat{x} - x}{x} \right|,
\]

and if \( \hat{x} \) and \( x \) are close:

**Example** \( f(x) = 2x \)

**Example** \( f(x) = \sin(2^{60}x) \).

---

**Definition**

**forward error analysis**

How much error do I have in solving problem \( f(x) \) using algorithm \( \text{alg} \)?

\[
\frac{|\text{alg}(x) - f(x)|}{f(x)}
\]

**backward error analysis**

Does my algorithm solve a nearby problem exactly?

\[
\text{alg}(x) = f(x + \delta x)
\]
**Backwards Stable Algorithms**

1. inner products
2. solving $Ax = b$
3. computing $QR$

**Matrix Norms**

$|\alpha| = \max |\alpha x|$ where $|x| = 1$

*how much it scales a unit quantity*

$\|A\|_2 = \text{how much it scales a unit vector}$

$= \max \|Ax\|_2$ where $\|x\| = 1.$

$= \sqrt{\text{largest eigenvalues of } A^T A}.$

**Conditioning of Linear Systems**

$Ax = b \quad b \approx \hat{b} \quad \hat{A}x = \hat{b}$

$$\frac{|x - \hat{x}|}{\|x\|} =$$

*in Julia: cond(A)*