\[ \min_{x \in \mathbb{R}^n} f(x) \]

**Gradient Alg.**

\[ x \leftarrow x - t \nabla f(x)^T \]

\# of iterations \( \sim \log \frac{1}{\epsilon} \)

**Newton's method**

\[ x \leftarrow x - t H(f(x))^{-1} \nabla f(x)^T \]

\# of iterations: \( \log \log \frac{1}{\epsilon} \) if the initial value is close enough to a local solution.

---

- Which algorithm is better?

  - many iterations for Gradient alg. \( \iff \) modest number of iterations for Newton's
  
  \[ \text{Example: } 10000 \iff \text{Example: } 20 \]

  - cheap iterations \( \iff \) expensive iterations.

- Example: Imagine that \( x \in \mathbb{R}^{1000} \)

  \[ \Rightarrow H(f(x)) \text{ has } 10^6 \text{ entries and we need to take its inverse at each iteration.} \]

- we can compute \( H(f(x))^{-1} \nabla f(x)^T \) without explicitly computing \( H(f(x))^{-1} \), but it's still expensive.
Summary:

Gradient algorithm needs many cheap iterations to get a tolerance ε

→ Newton's method needs a few expensive iterations to get to a tolerance ε.

- How to evaluate the convergence? Draw a linear-log plot where the horizontal axis shows iterations and the vertical axis shows $f(x)$.

- Can we design an algorithm with a convergence rate better than Gradient and Newton's methods?

- Recall that $\nabla f(x^{(k)}) \Delta x^{(k)}$ accounts for the reduction at every iteration, and that's why we look for a descent direction: $\nabla f(x^{(k)}) \Delta x^{(k)} < 0$.

- What if we optimize the direction $\Delta x$?
\[ \min_{\Delta x^{(k)}} \nabla f(x^{(k)}) \Delta x^{(k)} \Rightarrow \text{looks plausible but} \]
\[ \text{the minimum would be } -\infty \text{ corresponding to a } \Delta x^{(k)} \text{ with unbounded entries.} \]

Recall that \( \Delta x^{(k)} \) accounts for direction and \( t^{(k)} \) scales it down or up. So, why not normalize \( \Delta x^{(k)} \)?

Consider an arbitrary norm \( \| \cdot \| \).

Example: \( \| \begin{bmatrix} 1 & -1 \\ 3 & 1 \end{bmatrix} \| = \sqrt{1^2 + (-1)^2 + 3^2} \)
\[ \Rightarrow \| 111 \pm 11 - 11 + 131 \| = \text{max} \left( \| 111 \pm 11 - 11 + 131 \| \right) \]
\[ : \text{different types of norms.} \]

\[ \Rightarrow \min_{\Delta x^{(k)}} \nabla f(x^{(k)}) \Delta x^{(k)} \quad \text{s.t.} \quad \| \Delta x^{(k)} \| = 1 \]

This is called steepest descent algorithm.

If \( \| \cdot \| \) = length of vector \( \Rightarrow \Delta x^{(k)} = -\frac{\nabla f(x^{(k)})}{\| \nabla f(x^{(k)}) \|} \)
\[ \downarrow \]
\[ \text{Normalized Gradient} \]
\[ \downarrow \]
\[ \text{Gradient method.} \]

If \( \| y \| = y^T m y \) for some fixed matrix \( m \),
\[ \Rightarrow \text{The optimal } \Delta x^{(k)} \text{ corresponds to Newton's method.} \]
Steepest descent includes Gradient and Newton's method.

Assume we use one of the previous algorithms and generates a sequence:

\[ x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow \ldots \rightarrow x^{(k)} \rightarrow \ldots \]

Question: where does it converge to (if any)?
Local min or saddle point.

Example:

If \( x^{(0)} < 0 \) \( \Rightarrow \) it converges to a minimum
If \( x^{(0)} \geq 0 \) \( \Rightarrow \) it converges to the saddle point \( x = 0 \)
Theorem: Consider a local minimum $x^*$. If $x^{(0)}$ is close enough to $x^*$, the sequence converges to $x^*$.

- I don't know the solution, how can I figure out my initial guess $x^{(0)}$ is good enough?
- Can we come up with a classes of functions for which $x^{(0)}$ doesn't matter?
- Yes, this is about convex optimization.

A function $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if for every two points $x$ and $y$, the line connecting $(x,f(x))$ to $(y,f(y))$ is above the function.
Mathematical definition:

\( f(\cdot) : \mathbb{R}^n \to \mathbb{R} \) is convex if for every \( x, y \in \mathbb{R}^n \) and \( \alpha \in [0, 1] \), we have:

\[
f(\alpha x + (1-\alpha) y) \leq \alpha f(x) + (1-\alpha) f(y)
\]

Illustration:

Examples of convex functions in \( \mathbb{R} \):

\[ x^2 \]

\[ e^x \]

\[ x^4 \]

\[ e^{-x} \]