Review

A general descent method: \( x_{k+1} = x_k + \alpha_k p_k \)

\( p_k \) is a **descent direction** if \( p_k^T \nabla f(x_k) < 0 \).

The **line search** subproblem: \( \min_{\alpha > 0} f(x_k + \alpha p_k) \):

- Can be solved exactly for quadratic \( f \).
- For general \( f \), find “good enough” \( \alpha \), not too large or small

Too large \( \alpha \) may cause \( f(x_k + \alpha p_k) > f(x_k) \! \)

Not enough to demand \( f(x_k + \alpha p_k) < f(x_k) \) (to prove convergence).

Need the stronger **Armijo condition**:

\[
 f(x_k + \alpha p_k) \leq f(x_k) + \alpha \mu p_k^T \nabla f(x_k)
\]

Very small \( \mu = 10^{-4} \) OK.
Convergence with Newton’s method only if starting close to solution.

The Newton direction \( s_k = - (\nabla^2 f(x_k))^{-1} \nabla f(x) \) is a descent direction \( (s_k^T \nabla f(x_k) < 0) \) if the Hessian \( \nabla^2 f(x_k) \) is positive definite.

Line search with Newton’s method is called **Damped Newton Backtracking:**

- Start with \( \alpha = 1 \)
- Check the Armijo condition
- If needed, reduce \( \alpha \) until the Armijo condition is satisfied.
- \( \alpha < 1 \) only needed when far from the solution. When close enough, Newton will converge without step size reduction.
Damped Newton with modified Cholesky

1. Choose initial guess $x_0$, tolerance $\epsilon$, and parameter $\mu$

2. For $k = 0, 1, \ldots$
   (a) If $\|\nabla f(x_k)\| \leq \epsilon$ stop
   (b) Compute modified factorization $LL^T$ of $\nabla^2 f(x_k)$
       ($LL^T = E + \nabla^2 f(x_k)$ for some diagonal $E \geq 0$)
   (c) Solve $LL^T s_k = -\nabla f(x_k)$
   (d) $\alpha_k \leftarrow 1$
   (e) While $f(x_k + \alpha_k s_k) > f(x_k) + \mu \alpha_k s_k^T \nabla f(x_k)$,
       $\alpha \leftarrow \alpha/2$
   (f) Set $x_{k+1} \leftarrow x_k + \alpha s_k$
The steepest-descent method

Computing the Newton step with the crude approximation
\[ \nabla^2 f(x_k) \approx I \] yields
\[ s_k = -\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k) \approx -\nabla f(x_k) \]

\[ \nabla f(x_k) \]

\[ x_k \]

\[ -\nabla f(x_k) \]

The negative gradient:
- Is perpendicular to the level curves
- Points in the steepest downhill direction

The steepest-descent direction is the best direction locally
The steepest-descent algorithm:

- Easy to implement
- Does not need second derivatives: useable for large problems
- Only linear convergence rate, even with exact line search
- Terribly inefficient for ill-conditioned problem: large condition numbers of the Hessian
- Usually much better to use the conjugate gradient or a quasi-Newton algorithm