# Machine Learning and Optimization

## Lecture 10: Second-Order Methods and Globalization

CSCI 4140/6960 — Spring 2026

Course website: `cs.rpi.edu/~gittea/.../mlandopt.html`

# Learning Objectives

By the end of this lecture, you should be able to:

1. **Explain geometrically** why gradient descent is slow on ill-conditioned convex problems
2. **Derive Newton's method** as minimization of a local quadratic model
3. **State and interpret** local quadratic convergence of Newton's method, including the digits-doubling property and its basin of validity
4. **Explain what "globalization" means** and why it is needed even for convex optimization
5. **Distinguish** between different notions of "global convergence" in optimization

# Learning Objectives (continued)

6. **Motivate and explain** Armijo backtracking line search and its key properties
7. **Explain and compare** line search and trust-region globalization strategies for GD and Newton
8. **Describe and implement** a practical hybrid strategy that switches from GD to Newton

# Outline


- Motivation: Curvature is the Bottleneck
- Newton's Method: Local Quadratic Models
- Local Quadratic Convergence of Newton
- Why Newton is Not Globally Safe
- Why GD is Globally Safe: A Global Quadratic Model
- What is Globalization?
- Armijo Line Search: Motivation and Guarantees
- Algorithm: GD + Armijo Line Search
- Algorithm: Damped Newton + Line Search
- Trust-Region Newton: Motivation and Mechanics
- Algorithm: Trust-Region Newton
- Switching Strategies: GD $\rightarrow$ Newton
- Synthesis and Big Picture
- Appendix A: Proofs of Main Results
  - A.1: Newton's Local Quadratic Convergence
  - A.2: Armijo Backtracking Termination
  - A.3: Damped Newton Global-to-Local Convergence

# Assumptions

Throughout the lecture we will assume:

- **Base setting:** convex, $L$-smooth objective $f : \mathbb{R}^d \to \mathbb{R}$
- **Strong convexity** (when explicitly stated): $f$ is $\mu$-strongly convex
- **Newton local rates** additionally assume:
  - $f \in C^2$ (twice continuously differentiable)
  - Hessian $\nabla^2 f$ is Lipschitz with constant $M$ in a neighborhood of $x^\star$
  - $\nabla^2 f(x^\star) \succ 0$ (local strong convexity at the minimizer)

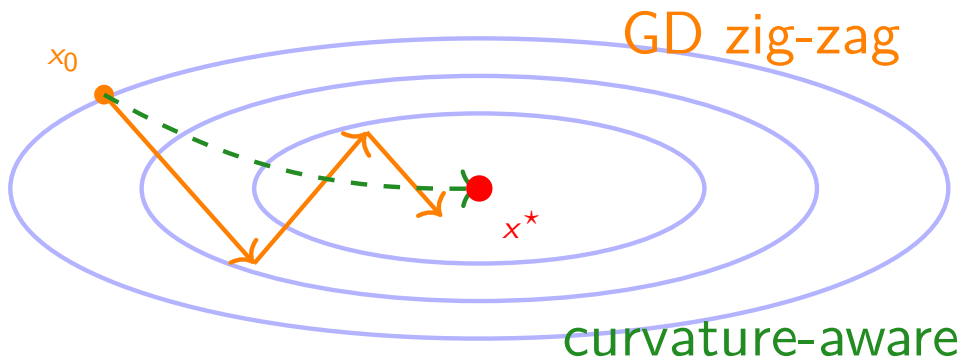
We will be explicit about which results require strong convexity vs convexity alone.

# Why Gradient Descent is Slow

Recall: for $\mu$-strongly convex and $L$-smooth $f$,

$$\|x_k - x^\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x_0 - x^\star\|_2, \quad \kappa = L/\mu.$$

- Large condition number $\kappa \Rightarrow$ slow linear convergence
- Geometry: elongated ellipses $\Rightarrow$ zig-zagging trajectories



# Motivating Question

*How can we use 2nd-order oracle information (curvature) to remove this geometric inefficiency?*

# Local Taylor Model

Answer: use a model that incorporates curvature!

$$m_x(\Delta) = f(x) + \langle \nabla f(x), \Delta \rangle + \tfrac{1}{2} \Delta^\top \nabla^2 f(x) \Delta.$$

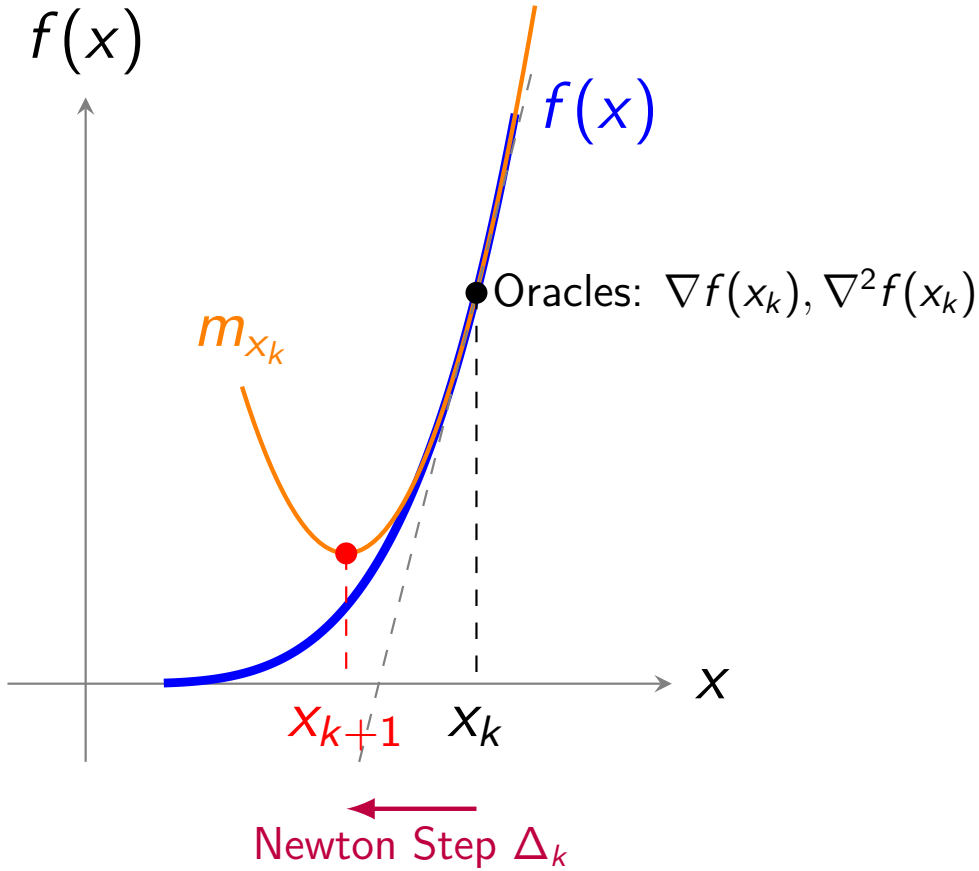
Newton step:

$$\Delta_k = \arg\min_{\Delta} m_{x_k}(\Delta)$$

$$\iff \quad \nabla^2 f(x_k) \Delta_k = -\nabla f(x_k).$$

**Interpretation:** Newton = exact minimization of the local quadratic model $\Rightarrow$ curvature-aware steps

# Geometric Picture



# Local Convergence of Newton

## Theorem (Local Quadratic Convergence)

*If $f$ is convex, $C^2$, $\nabla^2 f$ is Lipschitz with constant $M$ near $x^\star$, and $\nabla^2 f(x^\star) \succ \mu I$ (local strong convexity at the minimizer), then:*

$$\|x_{k+1} - x^\star\|_2 \leq \frac{M}{2\mu}\|x_k - x^\star\|_2^2$$

*for $x_k$ sufficiently close to $x^\star$.*

**Caution:** This is a **local** rate—only valid near the optimum.

# Basin of Attraction

Quadratic convergence only holds once

$$\|x_k - x^\star\|_2 \lesssim \frac{\mu}{M}.$$

Outside this basin, Newton behaves like a first-order method.

This motivates:

- **Globalization strategies** (to reach the basin)
- Hybrid methods (GD far away, Newton close in)

# Digits Double at each Iteration!

Quadratic convergence means:

$$\log_{10} \|x_{k+1} - x^\star\|_2 \approx 2 \log_{10} \|x_k - x^\star\|_2 + \text{const.}$$

**The number of digits of accuracy approximately doubles each iteration.**

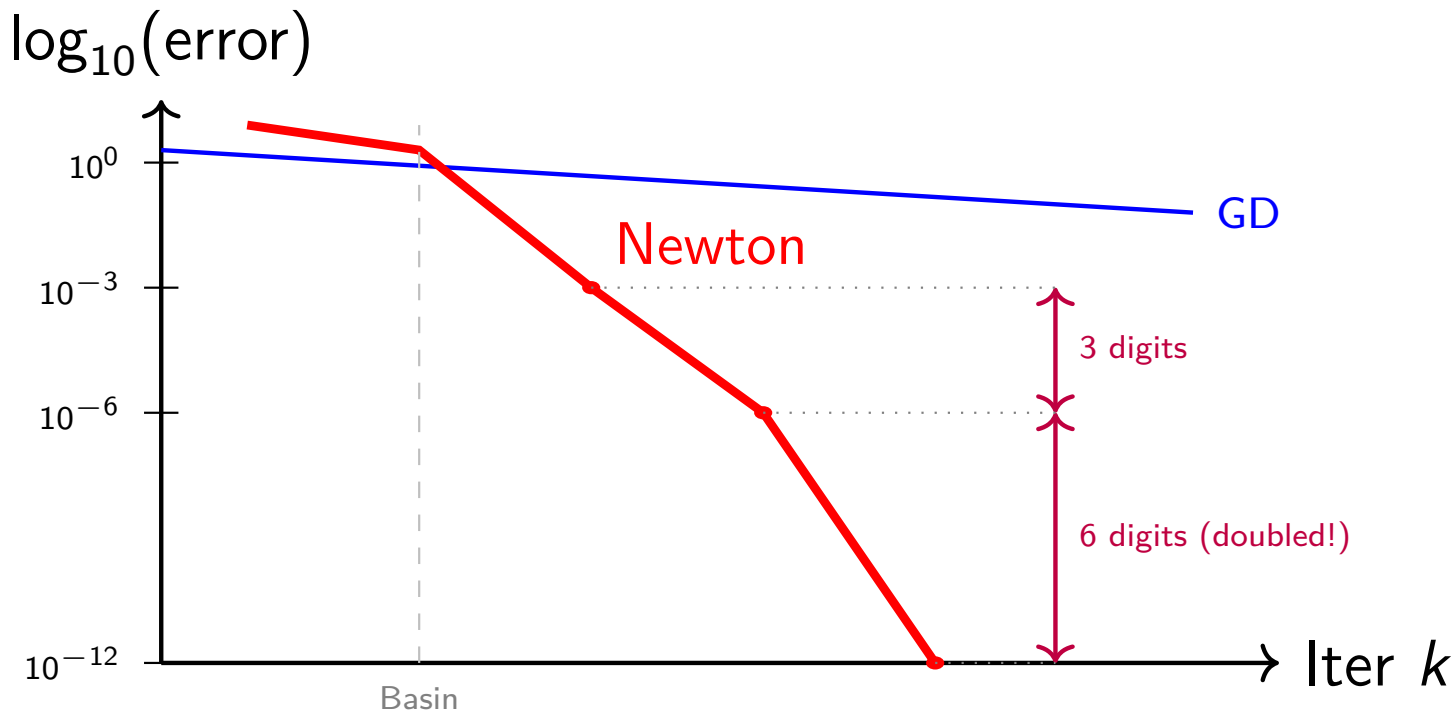
**Example:**

- If $\|x_k - x^\star\|_2 = 10^{-3}$, then $\|x_{k+1} - x^\star\|_2 \approx 10^{-6}$
- Then $\|x_{k+2} - x^\star\|_2 \approx 10^{-12}$

# Newton vs GD (locally)!

**For high-accuracy solutions** (e.g., $10^{-12}$ tolerance):

- Newton reaches target in $\sim$4 iterations *after entering the basin*
- GD with linear rate 0.9 needs $\sim 22$ *per digit* of accuracy, so $\sim 262$ iterations total.



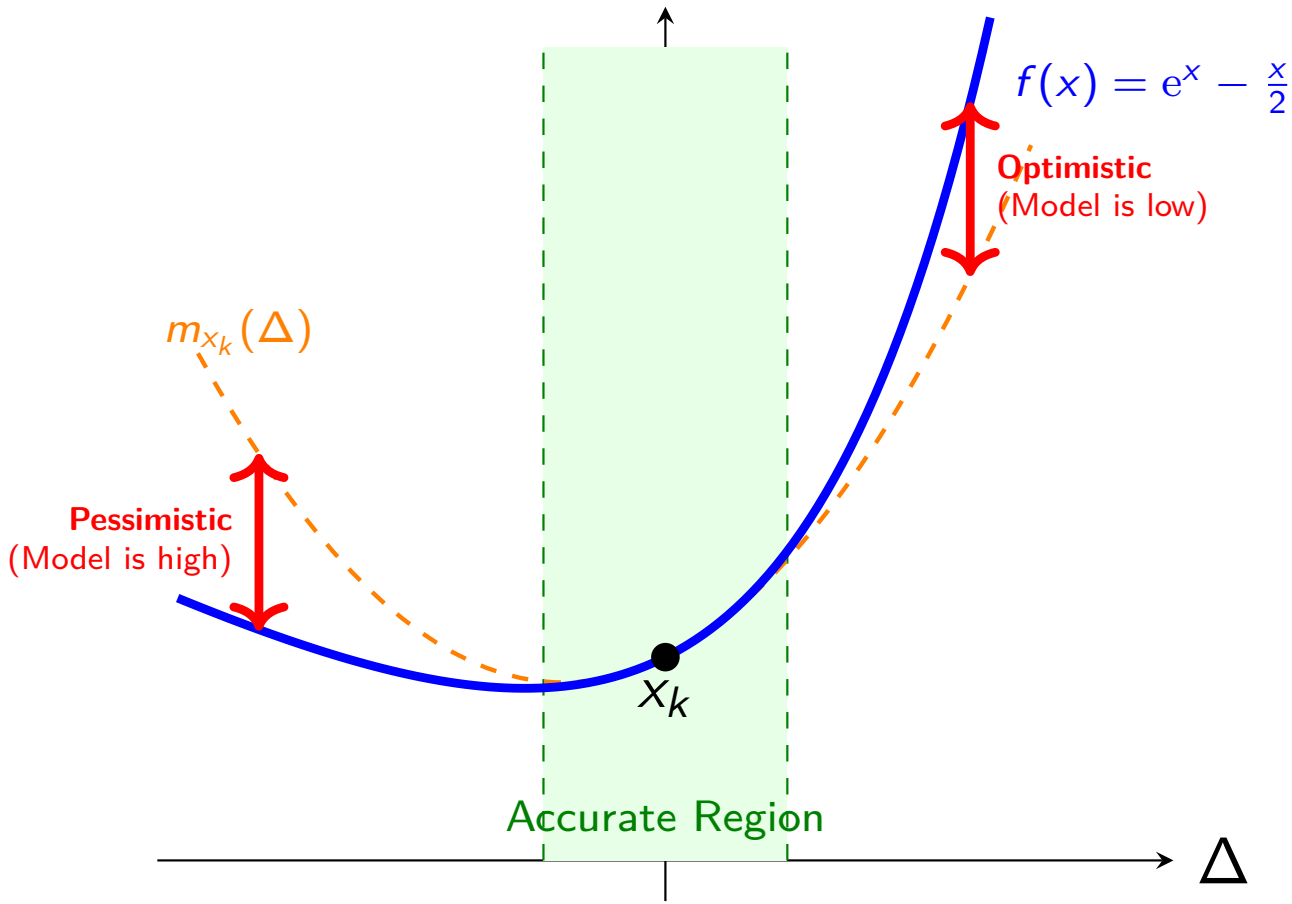
# How accurate is the local model?

If the Hessian is Lipschitz with constant $M$:

$$f(x+\Delta) \leq m_x(\Delta) + \frac{M}{6}|\Delta|^3.$$

- Quadratic model is only accurate locally (within $O(1/M)$ ball)
- **Convexity controls the objective, not the validity of the quadratic approximation**
- Far from $x^\star$, the cubic error term dominates and Newton can take non-descent steps or make extremely slow progress

# Accuracy: Convexity vs. Model Validity




Notice the red gaps: The model is only a "good" proxy inside the green zone.

**Takeaway:** Trusting the quadratic model too far leads to bad steps, even if the function is perfectly convex.

# Canonical Example

$$f(x) = x^4 + x^2$$

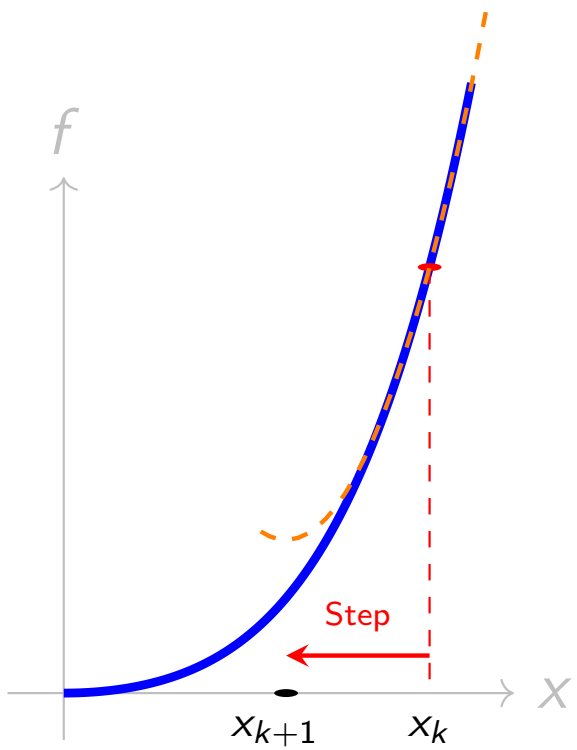
Newton update:

$$x_{k+1} = x_k - \frac{4x_k^3 + 2x_k}{12x_k^2 + 2}$$

Far from 0, the quartic dominates and Newton steps are tiny relative to distance.

**This failure happens even though $f$ is convex.**

**Salient question**: what makes *gradient descent* safe?



# Descent Lemma (Global Upper Bound)

**If $f$ is $L$-smooth**, then for all $x, \Delta$:

$$f(x+\Delta) \leq f(x) + \langle \nabla f(x), \Delta \rangle + \frac{L}{2}|\Delta|^2.$$

This upper bound is valid **everywhere**, not just locally.

GD with $\Delta = -\alpha \nabla f(x)$ minimizes the RHS over $\Delta$ (choosing $\alpha = 1/L$).

Thus the fixed step size $\alpha \leq 1/L$ **guarantees descent** from any starting point.

# Contrast with Newton

**Geometric Interpretation:** GD uses the conservative upper bound $LI$ on the curvate, valid everywhere for $L$-smooth functions

On the other hand, **Newton** uses the exact local curvature $\nabla^2 f(x)$, valid only near $x$.

GD trades speed for global safety. Newton trades global safety for speed. Note also Newton does not need to know or assume $L$-smoothness.

**How can we get the best of both worlds: speed and safety?**

# Introducing: Globalization

**Globalization** = modify locally motivated steps to ensure **global convergence** by controlling **model mismatch**.

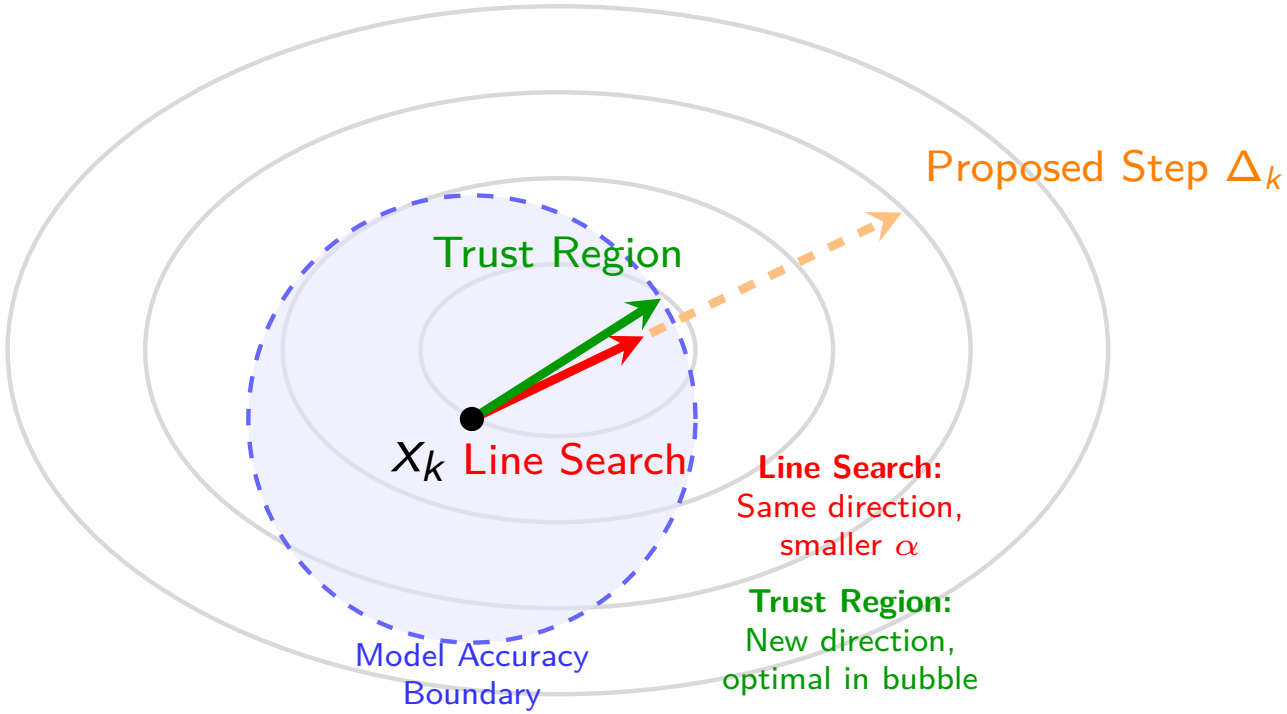
Two mechanisms:

- **Line search:** Accept the direction, but shrink the step size until model is accurate enough
- **Trust region:** Restrict optimization to a region where the model is trusted

The **cost** of globalization is extra oracle evaluations $f, \nabla f$ for line search, and constrained subproblem solves for trust regions

Globalization's benefits: **global convergence** from any starting point, and **fast local rates** near optima.

# Line Search vs Trust Region




**Line Search**
Checks points along the ray $x_k + \alpha\Delta_k$.

**Trust Region**
Solves $\min m_x(\Delta)$ s.t. $\|\Delta\| \leq r$.

# What "Global Convergence" Means

We need to distinguish three different notions:

| **Notion** | **Meaning** |
|---|---|
| Global descent | $f(x_{k+1}) \leq f(x_k)$ from any starting point |
| Global convergence to stationary point | $\|\nabla f(x_k)\| \to 0$ from any starting point |
| Global convergence to minimizer | $x_k \to x^\star$ from any starting point |

# What We Mean in This Lecture

**In this lecture:**

- All globalization methods ensure **monotone decrease** and convergence to stationary points from any starting point
- **Convexity** $\Rightarrow$ stationary points are global minimizers (so convergence to stationary point $=$ convergence to minimizer)
- **Strong convexity** $\Rightarrow$ unique minimizer $+$ quantitative convergence rates

These distinctions become crucial in nonconvex optimization (e.g., deep learning), where you can e.g. show convergence to stationary only.

# Line Search: What Makes a Good Step Size?

1. Sufficient descent (make progress)
2. Not too conservative (don't waste the directional information)
3. Use local model predictions, but verify against reality

# The Armijo Condition

For descent direction[1] $\Delta$ (i.e., $\langle \nabla f(x), \Delta \rangle < 0$):

$$f(x + \alpha\Delta) \leq f(x) + c\alpha\langle \nabla f(x), \Delta \rangle, \quad c \in (0, 1).$$

**Interpretation:**

$$\underbrace{f(x) - f(x + \alpha\Delta)}_{\text{actual decrease}} \geq \underbrace{-c\alpha\langle \nabla f(x), \Delta \rangle}_{\text{fraction of predicted decrease}}.$$

Actual decrease $\geq$ fixed fraction $c$ of linearized model prediction.

**Typical values:** $c = 10^{-5}$ (very lenient—accept steps with even 0.001% of predicted decrease) to $c = 10^{-1}$.

[1] **Are Newton directions descent directions?!**

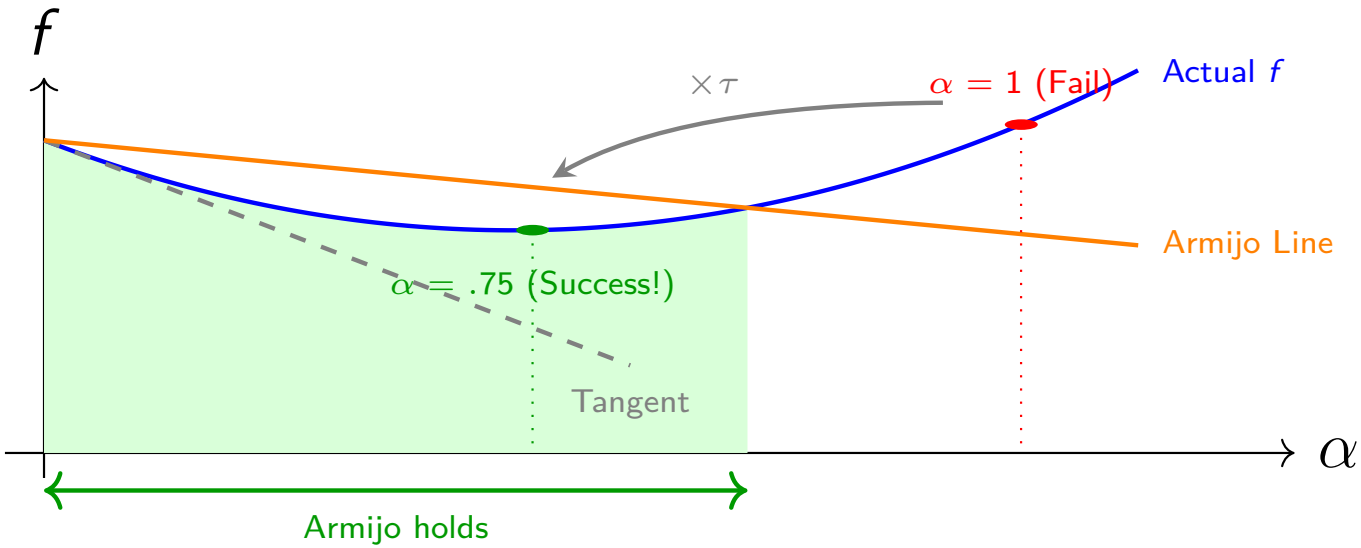
# What Armijo Actually Checks

**Critical:** Armijo checks the **first-order (linear) model**, not the quadratic model:

$$f(x + \alpha\Delta) \overset{?}{\approx} f(x) + \alpha\langle\nabla f(x), \Delta\rangle.$$

This means Newton + Armijo is globally safe even when the Hessian is inaccurate—we're only validating the gradient prediction, not the curvature prediction.

# Satisfying the Armijo condition via Backtracking




The initial step size $\alpha_{\text{init}} = 1$ is rejected because the objective value is significantly higher than the Armijo-acceptable threshold.

The step size is shrunk ($\alpha \leftarrow \tau\alpha$). The new point falls into the green zone, where Armijo holds

# Why Backtracking?

**Alternative:** Solve $\min_\alpha f(x + \alpha\Delta)$ "exactly"

**Problem:** Each exact line search (via binary search) costs $O(\log(1/\epsilon))$ function evaluations, expensive for $f : \mathbb{R}^d \to \mathbb{R}$ with large $d$

**Backtracking solution:**

- Start with $\alpha = \alpha_{\text{init}}$ (e.g., 1)
- Shrink by factor $\rho$ (e.g., 3/4) until Armijo holds
- Usually accepts in 1–3 trials: cheap and "good enough"

# Sanity Checks

## Proposition (Termination)

*If $f$ is L-smooth and $\Delta$ is a descent direction, backtracking Armijo always terminates in finite steps.*

*Proof idea:* For $\alpha$ small enough, $f(x + \alpha\Delta) \approx f(x) + \alpha\langle \nabla f(x), \Delta \rangle$ (first-order Taylor), which satisfies Armijo. □

## Proposition (Sufficient Decrease)

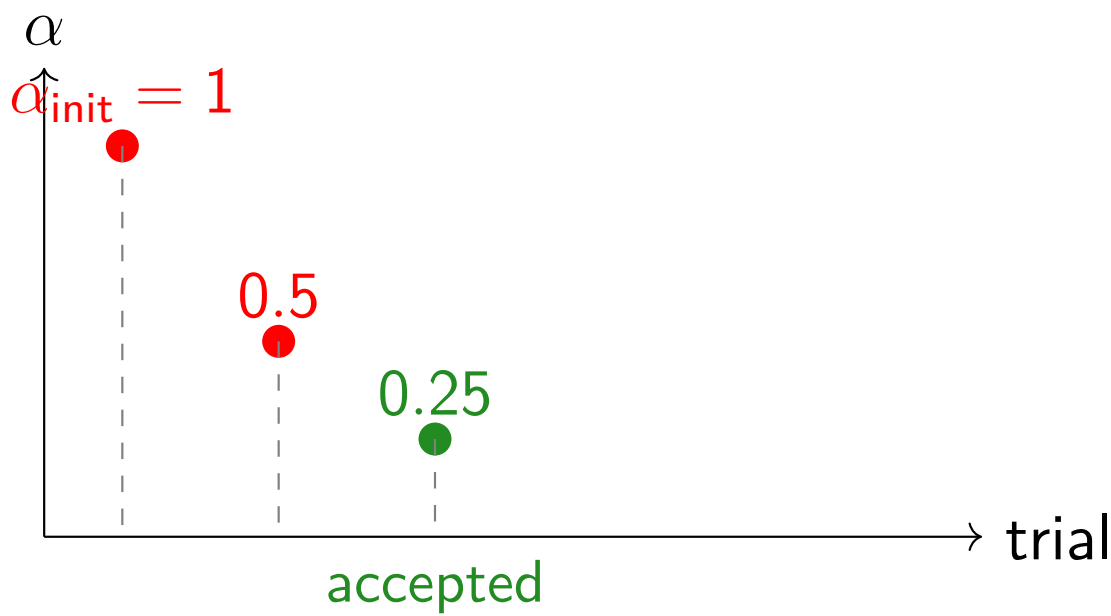
*For GD direction $\Delta = -\nabla f(x_k)$, accepted steps guarantee:*

$$f(x_{k+1}) \leq f(x_k) - c\alpha_k \|\nabla f(x_k)\|^2.$$

This is literally the Armijo condition.

# Takeaway

Armijo formalizes "take the biggest step the first-order model can justify" and guarantees sufficient decrease for GD.




Close to optimum, $\alpha = 1$ is usually accepted immediately.

# Backtracking Gradient Descent

Can be used when we don't know $L$, as opposed to the fixed step size $\alpha = \frac{1}{L}$.

---

**Algorithm 1:** Backtracking GD

---

**Input:** $x_0$, $c \in (0,1)$, $\tau \in (0,1)$, $\alpha_{\text{init}} > 0$
**for** $k = 0, 1, 2, \ldots$ **do**
    $\Delta_k \leftarrow -\nabla f(x_k)$;
    $\alpha \leftarrow \alpha_{\text{init}}$;
    **while** $f(x_k + \alpha\Delta_k) > f(x_k) + c\alpha\langle \nabla f(x_k), \Delta_k \rangle$ **do**
        $\alpha \leftarrow \tau\alpha$;
    **end**
    $x_{k+1} \leftarrow x_k + \alpha\Delta_k$;
**end**

---

**Guidance:** $c = 10^{-4}$, $\tau = 3/4$, $\alpha_{\text{init}} = 1$
**Cost per iteration:** $O(d)$ for gradient + $O(\log(L))$ function evaluations for line search

# Newton with Armijo Backtracking

**Algorithm 2:** Damped Newton (using backtracking with Armijo)

**Input:** $x_0$, $c, \tau, \alpha_{\text{init}}$

**for** $k = 0, 1, 2, \ldots$ **do**

  Solve $\nabla^2 f(x_k)\Delta_k = -\nabla f(x_k)$;

  $\alpha \leftarrow \alpha_{\text{init}}$;

  **while** $f(x_k + \alpha\Delta_k) > f(x_k) + c\alpha\langle\nabla f(x_k), \Delta_k\rangle$ **do**

    $\alpha \leftarrow \tau\alpha$;

  **end**

  $x_{k+1} \leftarrow x_k + \alpha\Delta_k$;

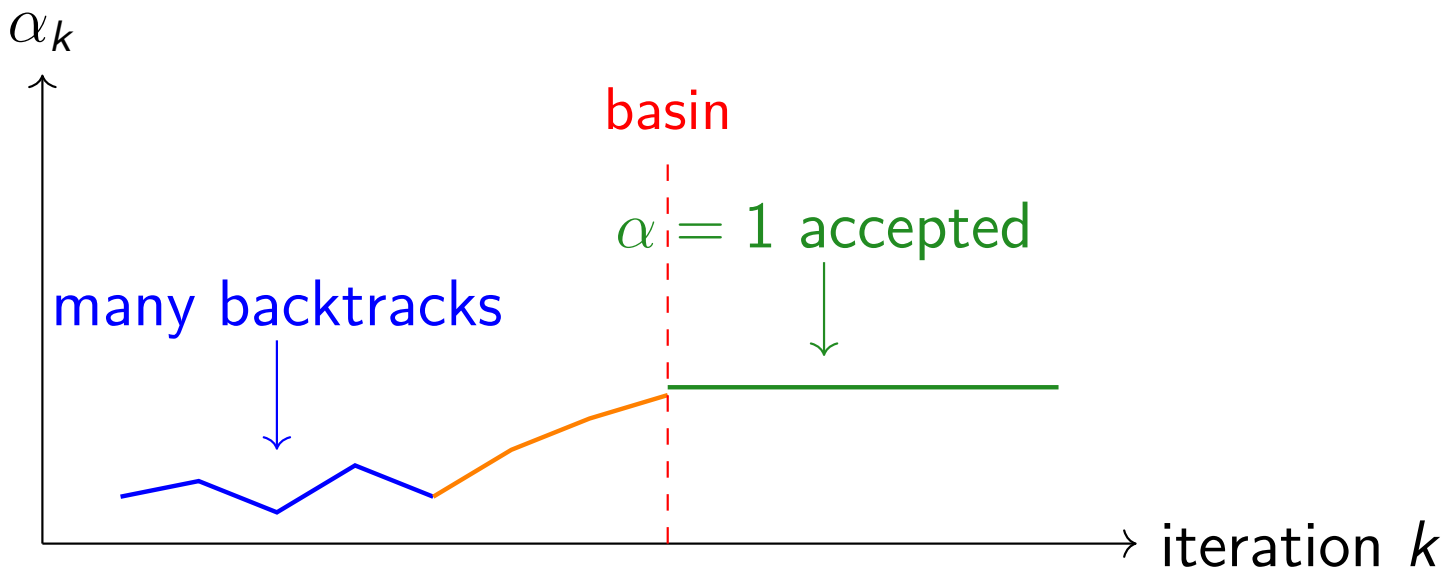
**end**

# Properties of Damped Newton

- **Globally convergent to stationary points** (under convexity + smoothness):
  Armijo ensures descent from any starting point
- **Eventually $\alpha_k = 1$ near $x^\star$:**
  Quadratic model becomes accurate, full Newton step satisfies Armijo
- **Recovers quadratic convergence locally:**
  Once $\alpha_k = 1$ accepted, pure Newton dynamics take over

**Cost per iteration:** $O(d^3)$ for dense Hessian factorization, or $O(d^2)$ per CG iteration for matrix-free Hessian-vector products + $O(\log(\frac{\mu}{M\|\Delta_k\|_2}))$ function evals for line search

# Step Size Behavior




Initially: many backtracks, $\alpha_k \ll 1$.
Near optimum: $\alpha_k = 1$ accepted immediately, marking entry into quadratic convergence regime.

# Trust-Region Subproblem

Instead of choosing the step size after choosing the direction, trust region methods globalize by simultaneously choosing the direction and magnitude:

$$\Delta_k = \arg \min_{\|\Delta\|_2 \leq r_k} \langle \nabla f(x_k), \Delta \rangle + \tfrac{1}{2} \Delta^\top \nabla^2 f(x_k) \Delta.$$

**Interpretation:**

- Optimize the quadratic model, but only within a"trust region" of radius $r_k$
- If the model is accurate, we trust it more so increase $r_k$
- If the model is inaccurate, we trust it less so decrease $r_k$

# Comparison to Line Search

| | **Line Search** | **Trust Region** |
|---|---|---|
| Direction | Fixed | Adaptive |
| Magnitude | Adaptive | Adaptive |
| Control | Step size $\alpha_k$ | Radius $r_k$ |

**Line search:** fixed direction, adaptive magnitude
**Trust region:** adaptive direction and magnitude together

# How to Solve the TR Subproblem (High-Level)

**Fermat's condition** implies the optimal $\Delta_k$ satisfies

$$\nabla m_x(\Delta_k) + \lambda \frac{\Delta_k}{\|\Delta_k\|_2} = 0$$

for some $\lambda \geq 0$ (**Show this!**).

Equivalently,

$$(\nabla^2 f(x_k) + \lambda I)\Delta_k = -\nabla f(x_k)$$

for some $\lambda \geq 0$.

- If unconstrained minimizer has norm $\leq r_k$: $\lambda = 0$, take a pure Newton step
- Otherwise: $\|\Delta_k\|_2 = r_k$ (solution on boundary) and $\lambda > 0$ chosen so $\|\Delta_k\|_2 = r_k$ exactly

# The Secular Equation

To find the boundary solution, we must solve for $\lambda > 0$ such that:

$$g(\lambda) = \|(\nabla^2 f(x_k) + \lambda I)^{-1} \nabla f(x_k)\|_2 = r_k$$

This is the so-called **secular equation**.

It has one unique solution $\lambda_k$, that can be solved for using a 1D Newton's method. Then we take

$$\Delta_k = (\nabla^2 f(x_k) + \lambda_k I)^{-1} \nabla f(x_k)$$

# Acceptance Ratio

After computing $\Delta_k$, we evaluate:

$$\rho_k = \frac{f(x_k) - f(x_k + \Delta_k)}{m_k(0) - m_k(\Delta_k)} = \frac{\text{actual decrease}}{\text{predicted decrease}}$$

to determine whether $r_k$ was a good radius (if model is accurately capturing curvature of $f$).

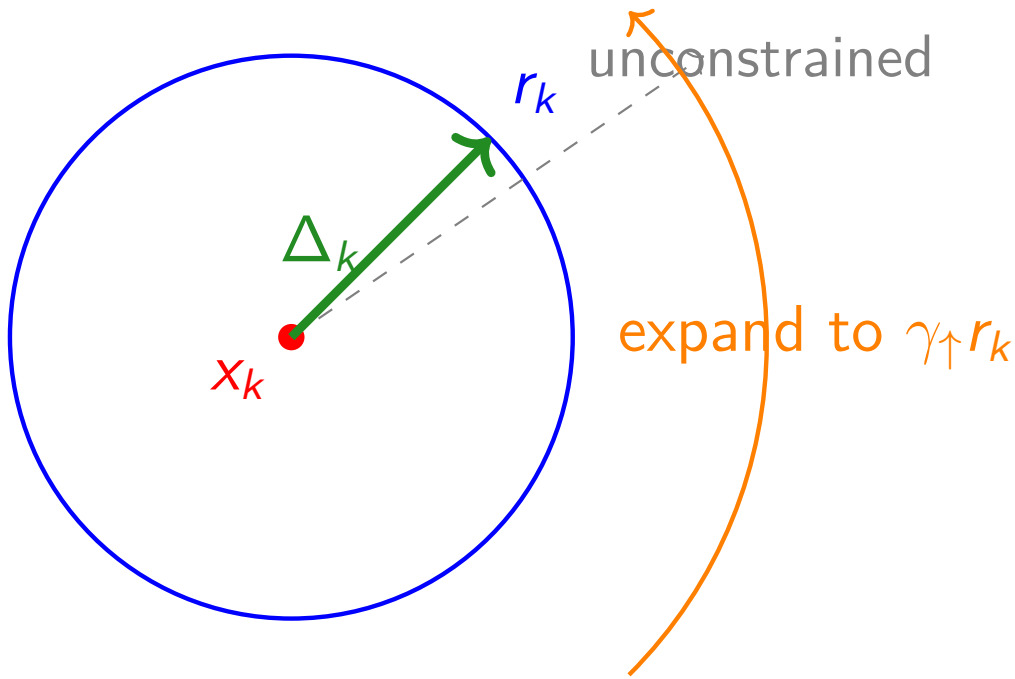
**Decision rule:**

- $\rho_k \geq \eta_1$ (e.g., 0.1): Radius is reasonable $\Rightarrow$ accept step
- $\rho_k < \eta_1$: Radius is too large $\Rightarrow$ reject step, shrink $r_k$
- $\rho_k > \eta_2$ (e.g., 0.9) **and** $\|\Delta_k\|_2 = r_k$:
  Model is excellent and we hit boundary $\Rightarrow$ expand $r_k$

# Why the Boundary Condition Matters

If $|\Delta_k| = r_k$, the optimizer wants to go farther but we stopped it at the trust radius.

If the model was accurate ($\rho$ high), we should trust it more and increase $r$.



# Trust-Region Newton Algorithm

**Algorithm 3:** Trust-Region Newton (Main Loop)

**Inputs:** $x_0, r_0 > 0, r_{\max}, 0 < \eta_1 < \eta_2 < 1, 0 < \gamma_\downarrow < 1 < \gamma_\uparrow$

**for** $k = 0, 1, 2, \ldots$ **do**

Compute $g_k = \nabla f(x_k)$, $H_k = \nabla^2 f(x_k)$;

Solve TR subproblem for $\Delta_k$;

$\text{pred}_k \leftarrow -\langle g_k, \Delta_k \rangle - \frac{1}{2}\Delta_k^\top H_k \Delta_k$;

$\text{ared}_k \leftarrow f(x_k) - f(x_k + \Delta_k)$;

$\rho_k \leftarrow \text{ared}_k / \text{pred}_k$;

**if** $\rho_k \geq \eta_1$ **then**

$x_{k+1} \leftarrow x_k + \Delta_k$ ; // accept

**else**

$x_{k+1} \leftarrow x_k$ ; // reject

**end**

$r_{k+1} \leftarrow \text{UpdateRadius}(\rho_k, \Delta_k, r_k)$ ; // see next slide

**end**

# Trust-Region Newton Algorithm (continued)

**Algorithm 4:** Trust-Region Subroutine

**Procedure** *UpdateRadius*$(\rho_k, \Delta_k, r_k)$ **is**

  **if** $\rho_k < \eta_1$ **then**

    **return** $\gamma_\downarrow r_k$ ; // shrink

  **else if** $\rho_k > \eta_2$ ***and*** $\|\Delta_k\|_2 = r_k$ **then**

    **return** $\min\{\gamma_\uparrow r_k, r_{\max}\}$ ; // expand

  **else**

    **return** $r_k$ ; // maintain

  **end**

**end**

**Guidance:** $\eta_1 = 0.1$, $\eta_2 = 0.9$, $\gamma_\downarrow = 0.25$, $\gamma_\uparrow = 2$, $r_{\max} = 10^{10}$

# Intuition

Trust-region methods naturally interpolate between GD-like (small $r$ adds large $\lambda$ regularization) and Newton-like (large $r$ makes $\lambda \approx 0$) behavior, without explicit switching.

**Cost per iteration:**

- $O(d^3)$ for Hessian (if computed densely)
- $O(d^2)$ to $O(d^3)$ for TR subproblem (depends on solver; Steihaug-CG is $O(d^2)$ per iteration)
- $O(d)$ for function/gradient eval

# Alternative: Switch from GD to Newton, explicitly

TR Newton interpolates between GD- and Newton-like behaviors, but *always* pays the cost of solving Hessian systems on each iteration.

Instead when far from the optimum use GD, which has linear convergence, and switch to Newton once inside the basin of attraction.

The difficulty is knowing when to switch algorithms.

# When to Switch?

**Signal:** Gradient norm small $\Rightarrow$ near optimum $\Rightarrow$ quadratic model accurate

**Why $\|\nabla f(x)\|_2$ works as a switching criterion (under strong convexity):**

- Small gradient implies proximity to $x^\star$
- In this regime, quadratic model error $O(\|x - x^\star\|_2^3)$ is negligible

# Critical Caveat

This criterion relies on **local strong convexity**.

On flat objectives (e.g., plateaus in deep learning), small gradient does **not** imply closeness to optimum.

This is one reason why second-order methods behave differently in nonconvex ML settings (saddle points, flat minima, etc.).

**Alternative criterion:** switch when line search consistently accepts $\alpha \approx 1$

# Hybrid Strategy

**Algorithm 5:** GD $\rightarrow$ Newton Switch

**Inputs:** $x_0$, tolerance $\varepsilon_{\text{switch}}$, final tolerance $\varepsilon_{\text{final}}$

`// Phase 1: GD`

**while** $\|\nabla f(x_k)\|_2 > \varepsilon_{switch}$ **do**

  Run GD + Armijo line search;

**end**

`// Phase 2: Newton`

**while** $\|\nabla f(x_k)\|_2 > \varepsilon_{final}$ **do**

  Run damped Newton + Armijo;

**end**

**Typical values:** $\varepsilon_{\text{switch}} = 10^{-2}$, $\varepsilon_{\text{final}} = 10^{-12}$

# The Core Tradeoff

| **Method** | **Model** | **Validity** | **Convergence** | **Cost/iter** |
|---|---|---|---|---|
| GD | Global quad. upper bound | Everywhere | $O(1/k)$ or geometric | $O(d)$ |
| Newton | Local quad. (exact $H$) | Inside $O(\mu/M)$ ball | Quadratic (digits double) | $O(d^3)$ dense $O(d^2)$/CG |

# Three Takeways

**1. GD is globally safe because it optimizes a global upper model.**

- Conservative curvature estimate $L I$ works everywhere
- The descent lemma provides a global quadratic upper bound
- Pays for global safety with slow convergence

# Three Core Lessons (continued)

**2. Newton is locally fast because it optimizes an exact local quadratic model.**

- Uses true Hessian: captures geometry exactly
- Achieves digits-doubling: critical for high-accuracy solutions
- Pays for speed with local-only validity (requires $|x - x^\star| \lesssim \mu/M$)

# Three Core Lessons (continued)

3. **Globalization is how we get the best of both worlds.**
   - Line search: shrink step size when first-order model fails
   - Trust region: shrink trusted region when quadratic model fails
   - Both recover local quadratic convergence near optimum

# Practical Takeaway

For solving $\min f(x)$ to high accuracy:

- **Low-to-moderate accuracy** ($\varepsilon \sim 10^{-3}$):
  GD often sufficient, cheaper overall
- **High accuracy** ($\varepsilon \sim 10^{-12}$):
  Newton essential—the digits-doubling property dominates cost in the final iterations
- **Best practice for general convex:**
  Trust-region Newton (robust, automatic interpolation between GD and Newton)
- **Simplest to implement:**
  GD $\rightarrow$ Newton hybrid with gradient-norm switching

# Questions?

Next time: Accelerated first-order methods

# Appendix A: Proofs of Main Results

This appendix provides detailed proofs of the main theoretical results from the lecture:

1. Newton's Local Quadratic Convergence Theorem
2. Armijo Backtracking Termination
3. Damped Newton Global-to-Local Convergence

# Theorem Statement (Reminder)

## Theorem (Local Quadratic Convergence)

*Suppose $f : \mathbb{R}^d \to \mathbb{R}$ satisfies:*

- *$f$ is convex and $C^2$*
- *$\nabla^2 f$ is Lipschitz continuous with constant $M$ in a neighborhood of $x^\star$*
- *$\nabla^2 f(x^\star) \succ \mu I$ (local strong convexity at the minimizer)*

*Then for $x_0$ sufficiently close to $x^\star$, Newton's method satisfies:*

$$\|x_{k+1} - x^\star\|_2 \leq \frac{M}{2\mu}\|x_k - x^\star\|_2^2$$

**Roadmap**: We'll prove this in 4 steps.

# Step 1: Taylor Expansion of the Gradient

At the optimum, $\nabla f(x^\star) = 0$. For any $x_k$, we can write:

$$\nabla f(x_k) = \nabla f(x^\star) + \int_0^1 \nabla^2 f(x^\star + t(x_k - x^\star))(x_k - x^\star)\, dt$$

Using $\nabla f(x^\star) = 0$:

$$\nabla f(x_k) = \int_0^1 \nabla^2 f(x^\star + t(x_k - x^\star))\, dt \cdot (x_k - x^\star)$$

Split the integral:

$$\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + R_k$$

where $R_k = \int_0^1 [\nabla^2 f(x^\star + t(x_k - x^\star)) - \nabla^2 f(x^\star)](x_k - x^\star)\, dt$

# Step 2: Bounding the Remainder

We need to bound $\|R_k\|_2$. Using the Hessian Lipschitz condition:

$$\begin{aligned}
\|R_k\|_2 &= \left\| \int_0^1 [\nabla^2 f(x^\star + t(x_k - x^\star)) - \nabla^2 f(x^\star)](x_k - x^\star)\, dt \right\|_2 \\
&\leq \int_0^1 \|\nabla^2 f(x^\star + t(x_k - x^\star)) - \nabla^2 f(x^\star)\|_2 \|x_k - x^\star\|_2\, dt \\
&\leq \int_0^1 Mt\|x_k - x^\star\|_2 \cdot \|x_k - x^\star\|_2\, dt \\
&= M\|x_k - x^\star\|_2^2 \int_0^1 t\, dt = \frac{M}{2}\|x_k - x^\star\|_2^2
\end{aligned}$$

# Step 3: Newton Step Analysis

The Newton iteration is:

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Subtract $x^\star$ from both sides:

$$x_{k+1} - x^\star = x_k - x^\star - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Substitute the Taylor expansion $\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + R_k$:

$$\begin{aligned} x_{k+1} - x^\star &= x_k - x^\star - (\nabla^2 f(x_k))^{-1} [\nabla^2 f(x^\star)(x_k - x^\star) + R_k] \\ &= x_k - x^\star - (\nabla^2 f(x_k))^{-1} \nabla^2 f(x^\star)(x_k - x^\star) \\ &\quad - (\nabla^2 f(x_k))^{-1} R_k \end{aligned}$$

# Step 3: Newton Step Analysis (continued)

Factor out $(\nabla^2 f(x_k))^{-1}$:

$$\begin{aligned} x_{k+1} - x^\star &= (\nabla^2 f(x_k))^{-1}[\nabla^2 f(x_k)(x_k - x^\star) - \\ &\qquad \nabla^2 f(x^\star)(x_k - x^\star) - R_k] \\ &= (\nabla^2 f(x_k))^{-1}[(\nabla^2 f(x_k) - \nabla^2 f(x^\star))(x_k - x^\star) - R_k] \end{aligned}$$

Taking norms:

$$\begin{aligned} \|x_{k+1} - x^\star\|_2 \leq \|(\nabla^2 f(x_k))^{-1}\|_2 \cdot \|\nabla^2 f(x_k) - \nabla^2 f(x^\star)\|_2 \|x_k - x^\star\|_2 \\ + \|(\nabla^2 f(x_k))^{-1}\|_2 \|R_k\|_2 \end{aligned}$$

# Step 4: Applying Strong Convexity and Lipschitz Bounds

We need to bound two quantities:

**1. Inverse Hessian norm:**
From local strong convexity $\nabla^2 f(x) \succ \mu I$ near $x^\star$:

$$\|(\nabla^2 f(x_k))^{-1}\|_2 \leq \frac{1}{\mu}$$

**2. Hessian difference:**
From Hessian Lipschitz continuity:

$$\|\nabla^2 f(x_k) - \nabla^2 f(x^\star)\|_2 \leq M\|x_k - x^\star\|_2$$

# Step 4: Final Assembly

Combining all bounds:

$$\begin{aligned}\|x_{k+1} - x^\star\|_2 &\leq \frac{1}{\mu} \cdot M\|x_k - x^\star\|_2 \cdot \|x_k - x^\star\|_2 + \frac{1}{\mu} \cdot \frac{M}{2}\|x_k - x^\star\|_2^2 \\ &= \frac{M}{\mu}\|x_k - x^\star\|_2^2 + \frac{M}{2\mu}\|x_k - x^\star\|_2^2 \\ &= \frac{3M}{2\mu}\|x_k - x^\star\|_2^2\end{aligned}$$

With more careful analysis (tighter bounds on the integral), one can improve the constant to:

$$\|x_{k+1} - x^\star\|_2 \leq \frac{M}{2\mu}\|x_k - x^\star\|_2^2 \qquad \square$$

# Conceptual Recap of the Proof

- The gradient $\nabla f(x_k)$ contains the "error" $\nabla^2 f(x^\star)(x_k - x^\star)$
- The Newton step $-(\nabla^2 f(x_k))^{-1}\nabla f(x_k)$ cancels this exactly when $\nabla^2 f(x_k) = \nabla^2 f(x^\star)$
- The remaining error comes from the Hessian mismatch, which is $O(\|x_k - x^\star\|)$
- Multiplying by the distance gives the quadratic term: $O(\|x_k - x^\star\|) \times O(\|x_k - x^\star\|) = O(\|x_k - x^\star\|^2)$

# Theorem Statement

## Theorem (Armijo Termination)

*Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and let $\Delta$ be a descent direction (i.e., $\langle \nabla f(x), \Delta \rangle < 0$).*
*Then backtracking line search with parameters $c \in (0, 1)$ and $\tau \in (0, 1)$ terminates in finite iterations with step size:*

$$\alpha \geq \min \left\{ 1, \frac{2\tau(1-c)}{L} \cdot \frac{-\langle \nabla f(x), \Delta \rangle}{\|\Delta\|_2^2} \right\}$$

**Interpretation**: The accepted step size has a positive lower bound that depends on $L$, $c$, $\tau$, and the descent quality.

# Proof: Step 1 - Descent Lemma

Recall the descent lemma (global quadratic upper bound for $L$-smooth functions):

$$f(x + \alpha\Delta) \leq f(x) + \alpha\langle \nabla f(x), \Delta \rangle + \frac{L\alpha^2}{2}\|\Delta\|_2^2$$

The Armijo condition requires:

$$f(x + \alpha\Delta) \leq f(x) + c\alpha\langle \nabla f(x), \Delta \rangle$$

Thus, if the RHS of the descent lemma is $\leq$ the RHS of Armijo, then Armijo is satisfied.

# Proof: Step 2 - Finding Compatible Step Sizes

We need:

$$f(x) + \alpha\langle\nabla f(x), \Delta\rangle + \frac{L\alpha^2}{2}\|\Delta\|_2^2 \leq f(x) + c\alpha\langle\nabla f(x), \Delta\rangle$$

Cancel $f(x)$:

$$\alpha\langle\nabla f(x), \Delta\rangle + \frac{L\alpha^2}{2}\|\Delta\|_2^2 \leq c\alpha\langle\nabla f(x), \Delta\rangle$$

Divide by $\alpha$ (positive):

$$\langle\nabla f(x), \Delta\rangle + \frac{L\alpha}{2}\|\Delta\|_2^2 \leq c\langle\nabla f(x), \Delta\rangle$$

Rearrange:

$$\frac{L\alpha}{2}\|\Delta\|_2^2 \leq (c-1)\langle\nabla f(x), \Delta\rangle$$

# Proof: Step 3 - Using the Descent Condition

Since $\Delta$ is a descent direction: $\langle \nabla f(x), \Delta \rangle < 0$
Therefore $(c-1)\langle \nabla f(x), \Delta \rangle > 0$ (since $c < 1$)

Solve for $\alpha$:

$$\alpha \leq \frac{2(c-1)\langle \nabla f(x), \Delta \rangle}{L\|\Delta\|_2^2}$$

Since $\langle \nabla f(x), \Delta \rangle < 0$, we can write this as:

$$\alpha \leq \frac{2(1-c)(-\langle \nabla f(x), \Delta \rangle)}{L\|\Delta\|_2^2}$$

Let $\bar{\alpha} = \frac{2(1-c)(-\langle \nabla f(x), \Delta \rangle)}{L\|\Delta\|_2^2}$.
Then any $\alpha \leq \bar{\alpha}$ satisfies Armijo.

# Proof: Step 4 - Backtracking Guarantee

Backtracking starts with $\alpha_0$ (typically 1) and shrinks by factor $\tau$ until Armijo holds.

**Observation**: Backtracking can "overshoot" $\bar{\alpha}$ by at most one factor of $\tau$.

Specifically, if $\alpha_{\text{accepted}}$ is returned by backtracking, then the previous (rejected) step size satisfies:

$$\alpha_{\text{rejected}} = \frac{\alpha_{\text{accepted}}}{\tau},$$

and since $\alpha_{\text{rejected}}$ was rejected, it must violate Armijo, so $\alpha_{\text{rejected}} > \bar{\alpha}$. This implies $\alpha_{\text{accepted}} > \tau\bar{\alpha}$.

Thus, indeed

$$\alpha_{\text{accepted}} > \tau\bar{\alpha} = \frac{2\tau(1-c)(-\langle \nabla f(x), \Delta \rangle)}{L\|\Delta\|_2^2}$$

# Proof: Step 5 - Final Bound

We also have the trivial bound: if $\alpha_0 = 1$ satisfies Armijo, then $\alpha_{\text{accepted}} = 1$.

Combining both cases:

$$\alpha_{\text{accepted}} \geq \min\left\{1, \frac{2\tau(1-c)(-\langle \nabla f(x), \Delta \rangle)}{L\|\Delta\|_2^2}\right\} \qquad \square$$

**Corollary**: For gradient descent direction $\Delta = -\nabla f(x)$:

$$-\langle \nabla f(x), \Delta \rangle = \|\nabla f(x)\|_2^2$$

$$\|\Delta\|_2^2 = \|\nabla f(x)\|_2^2$$

So:

$$\alpha_{\text{accepted}} \geq \min\left\{1, \frac{2\tau(1-c)}{L}\right\}$$

# Theorem: Global-to-Local Convergence of Damped Newton

## Theorem

*Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice continuously differentiable, $\mu$-strongly convex, L-smooth, and have M-Lipschitz Hessian. Newton's method with Armijo backtracking satisfies:*

1. ***(Global linear convergence – damped phase)*** *There exists $\rho \in (0, 1)$ such that for all $k$,*

$$f(x_{k+1}) - f(x^\star) \;\leq\; \rho\,(f(x_k) - f(x^\star)).$$

2. ***(Local quadratic convergence – pure Newton phase)*** *There exists $K$ such that for all $k \geq K$, Armijo accepts $\alpha_k = 1$ and*

$$\|x_{k+1} - x^\star\|_2 \;\leq\; \frac{M}{2\mu}\|x_k - x^\star\|_2^2.$$

# Phase 1: Damped Newton (Key Descent Inequality)

Armijo backtracking ensures:

$$f(x_{k+1}) \leq f(x_k) + c\alpha_k \langle \nabla f(x_k), \Delta_k \rangle$$

For the Newton direction $\Delta_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$,

$$\begin{aligned} \langle \nabla f(x_k), \Delta_k \rangle &= -\nabla f(x_k)^\top (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \\ &=: -\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}}. \end{aligned}$$

Therefore:

$$f(x_{k+1}) \leq f(x_k) - c\alpha_k \|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}}.$$

# Phase 1: Damped Newton (Uniform Step Size)

From the Armijo termination theorem, we have that

$$\alpha_k \geq \min\left\{1, \frac{2\tau(1-c)(-\langle \nabla f(x_k), \Delta_k \rangle}{L\|\Delta_k\|_2^2}\right\}$$

Let $v_k = \nabla^2 f(x_k)^{-1} \nabla f(x_k)$, then by the strong convexity of $f$,

$$\frac{-\langle \nabla f(x_k), \Delta_k \rangle}{\|\Delta_k\|_2^2} = \frac{v_k^T \nabla^2 f(x_k) v_k}{\|v_k\|_2^2} \geq \mu$$

It follows that

$$\alpha_k \geq \underline{\alpha} := \min\left\{1, \frac{2\tau(1-c)\mu}{L}\right\}$$

**Interpretation:** Backtracking never shrinks steps to zero. There is a fixed minimum step size during the damped phase.

# Phase 1: Damped Newton (From Decrement to Function Gap)

Since $f$ is $L$-smooth and $C^2$, we have:

$$\nabla^2 f(x_k) \preceq LI \quad \Rightarrow \quad (\nabla^2 f(x_k))^{-1} \succeq \frac{1}{L} I.$$

Therefore:

$$\|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \geq \frac{1}{L} \|\nabla f(x_k)\|_2^2.$$

By $\mu$-strong convexity, which implies the PL inequality,

$$\|\nabla f(x_k)\|_2^2 \geq 2\mu \big(f(x_k) - f(x^\star)\big).$$

# Phase 1: Damped Newton (Linear Convergence)

Combine the previous bounds:

$$\begin{aligned} f(x_{k+1}) - f(x^\star) &\le f(x_k) - f(x^\star) - c\alpha_k \|\nabla f(x_k)\|^2_{\nabla^2 f(x_k)^{-1}} \\ &\le f(x_k) - f(x^\star) - c\underline{\alpha} \cdot \frac{1}{L}\|\nabla f(x_k)\|_2^2 \\ &\le \left(1 - \frac{2c\,\underline{\alpha}\,\mu}{L}\right)\big(f(x_k) - f(x^\star)\big). \end{aligned}$$

Let

$$\rho := 1 - \frac{2c\,\underline{\alpha}\,\mu}{L} \in (0,1).$$

**Conclusion:** During the damped phase,

$$f(x_{k+1}) - f(x^\star) \le \rho\,(f(x_k) - f(x^\star)).$$

# Phase 2: Pure Newton = Quadratic Convergence

**Goal:** Once $x_k$ is close enough to $x^\star$, Armijo backtracking accepts

$$\alpha_k = 1,$$

and Newton enters the *pure Newton regime* with *quadratic convergence*:

$$\|x_{k+1} - x^\star\|_2 \leq C\|x_k - x^\star\|_2^2.$$

**Interpretation:** Global robustness first, local superlinear speed later.

# Phase 2: Local Model Accuracy (Lipschitz Hessian)

Because $\nabla^2 f$ is $M$-Lipschitz:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq M\|x - y\|_2.$$

Then the third-order Taylor remainder satisfies:

$$\left|f(x_k + \Delta) - \left[f(x_k) + \langle \nabla f(x_k), \Delta \rangle + \tfrac{1}{2}\Delta^T \nabla^2 f(x_k)\Delta\right]\right| \leq \frac{M}{6}\|\Delta\|_2^3.$$

**Interpretation:** Near $x^\star$, the quadratic model is accurate up to cubic error. This follows from a by now standard Taylor series argument.

# Phase 2: Full Newton Step Satisfies Armijo (Locally)

Apply the Taylor bound with the Newton step $\Delta_k = -(\nabla^2 f(x_k))^{-1}\nabla f(x_k)$:

$$f(x_k + \Delta_k) \leq f(x_k) + \langle \nabla f(x_k), \Delta_k \rangle + \tfrac{1}{2}\Delta_k^\top \nabla^2 f(x_k)\Delta_k + \frac{M}{6}\|\Delta_k\|_2^3$$

Since $\nabla f(x_k) = -\nabla^2 f(x_k)\Delta_k$:

$$\langle \nabla f(x_k), \Delta_k \rangle = -\Delta_k^\top \nabla^2 f(x_k)\Delta_k$$

Therefore:

$$f(x_k + \Delta_k) \leq f(x_k) - \tfrac{1}{2}\Delta_k^\top \nabla^2 f(x_k)\Delta_k + \frac{M}{6}\|\Delta_k\|_2^3.$$

# Phase 2: When Does $\alpha_k = 1$ Pass Armijo?

Armijo with $\alpha = 1$ requires:

$$f(x_k+\Delta_k) \le f(x_k)+c\langle \nabla f(x_k), \Delta_k\rangle = f(x_k)-c\Delta_k^\top \nabla^2 f(x_k)\Delta_k.$$

A sufficient condition is:

$$\left(\tfrac{1}{2}-c\right)\Delta_k^\top \nabla^2 f(x_k)\Delta_k \ge \frac{M}{6}\|\Delta_k\|_2^3.$$

Since $f$ is $\mu$-strongly convex, $\nabla^2 f(x_k) \succeq \mu I$:

$$\Delta_k^\top \nabla^2 f(x_k)\Delta_k \ge \mu\|\Delta_k\|_2^2.$$

So it suffices that:

$$\|\Delta_k\|_2 \le \frac{6(\frac{1}{2}-c)\mu}{M}.$$

# Phase 2: When Do Full Steps Start?

Relate the Newton step to the gradient:

$$\|\Delta_k\|_2 = \|(\nabla^2 f(x_k))^{-1}\nabla f(x_k)\|_2 \leq \frac{1}{\mu}\|\nabla f(x_k)\|_2.$$

Relate the gradient norm to the function gap (smoothness):

$$\|\nabla f(x_k)\|_2^2 \leq 2L\big(f(x_k) - f(x^\star)\big).$$

Therefore, a sufficient condition is:

$$f(x_k) - f(x^\star) \leq \frac{18(\frac{1}{2} - c)^2\,\mu^4}{L\,M^2}.$$

**Interpretation:** Once the function gap is below this explicit threshold, Armijo accepts $\alpha_k = 1$ and the algorithm switches to the pure Newton regime.

# Phase 2: Entering the Quadratic Basin (Sketch)

**Idea**: The global linear decrease in the objective continues at every iteration (whether steps are damped or full), so $f(x_k) \to f(x^\star)$ and hence $x_k \to x^\star$.

By $\mu$-strong convexity:

$$f(x_k) - f(x^\star) \;\geq\; \frac{\mu}{2}\|x_k - x^\star\|_2^2,$$

so linear decay of the objective gap implies $\|x_k - x^\star\|_2 \to 0$.

**Conclusion:** For any fixed radius $r > 0$, there exists $K$ such that

$$k \geq K \;\Rightarrow\; \|x_k - x^\star\|_2 \leq r.$$

# Phase 2: Once Full Steps Start, They Persist (Sketch)

Let $r > 0$ be a radius such that:

- if $\|x_k - x^\star\|_2 \le r$, then Armijo accepts $\alpha_k = 1$, and
- with $\alpha_k = 1$, Newton maps $B(x^\star, r)$ into itself.

Any $r \in \left(0, \min\{r_{\text{Armijo}}, \frac{2\mu}{M}\}\right)$ suffices, where $r_{\text{Armijo}}$ is chosen so that the sufficient condition for Armijo to take full steps is satisfied.

From the previous slide there exists $K$ such that $\|x_K - x^\star\|_2 \le r$

**Conclusion:** From iteration $K$ onward, $\alpha_k = 1$ forever and

$$\|x_{k+1} - x^\star\|_2 \le \frac{M}{2\mu}\|x_k - x^\star\|_2^2,$$

i.e., the algorithm takes full Newton steps and converges quadratically thereafter.