02610
Optimization and Data Fitting
Week 2: Line Search Methods

Yiqiu Dong

DTU Compute
Technical University of Denmark

Numerical optimization algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \qquad f:\mathbb{R}^n\to\mathbb{R},\ f\in\mathcal{C}^2(\mathbb{R}^n)$$

The iteration step in most algorithms that we will introduce in this course is essentially in the form of

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k.$$

- $\boldsymbol{x}_k$ is the iterate. We expect as $k\to+\infty$ we have $\boldsymbol{x}_k\to\boldsymbol{x}^*$.
- $\boldsymbol{p}_k$ is the search direction.
- α_k is the step size or step length.
- In addition, we need the stop criteria to define when the algorithm stops.

Numerical optimization algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \qquad f:\mathbb{R}^n\to\mathbb{R},\ f\in\mathcal{C}^2(\mathbb{R}^n)$$

The iteration step in most algorithms that we will introduce in this course is essentially in the form of

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k.$$

Line search Methods: The method chooses a direction $\boldsymbol{p}_k$ first, then searches along this direction from the current iterate $\boldsymbol{x}_k$ for a new iterate with a lower function value. For example, we can find the next step size by solving the univariate problem:

$$\min_{\alpha>0} f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k).$$

Search directions: Steepest descent direction

Goal:

$$f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f(\boldsymbol{x}_k)$$

Applying Taylor's theorem, we have

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}) = f(\boldsymbol{x}_k) + \alpha \boldsymbol{p}^T \nabla f(\boldsymbol{x}_k) + O(\alpha^2)$$

for any $\boldsymbol{p}$ and α. So the search direction for *descent methods* must satisfy

$$\boldsymbol{p}^T \nabla f(\boldsymbol{x}_k) < 0.$$

A normalized steepest descent direction is the solution of

$$\min_{\boldsymbol{p}} \boldsymbol{p}^T \nabla f(\boldsymbol{x}_k), \quad \text{subject to } \|\boldsymbol{p}\| = 1.$$

Steepest descent methods

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

where

$$\boldsymbol{p}_k = \arg\min_{\boldsymbol{p}} \boldsymbol{p}^T \nabla f(\boldsymbol{x}_k), \quad \text{subject to } \|\boldsymbol{p}\| = 1.$$

If we use 2-norm on the constraint, i.e., $\|\boldsymbol{p}\|_2 = 1$, then we have the closed-form solution

$$\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)/\|\nabla f(\boldsymbol{x}_k)\|_2.$$

NB: The methods using $-\nabla f(\boldsymbol{x}_k)$ as the search direction usually are called gradient descent methods, but in the textbook it is called THE steepest descent method.

Steepest descent methods

$$\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)/\|\nabla f(\boldsymbol{x}_k)\|_2.$$

- $\boldsymbol{p}_k$ is a descent direction:

$$\begin{aligned} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) &= -\alpha_k \boldsymbol{p}_k^T \nabla f(\boldsymbol{x}_k) + O(\alpha_k^2) \\ &= \alpha_k \|\nabla f(\boldsymbol{x}_k)\|_2 + O(\alpha_k^2) \end{aligned}$$

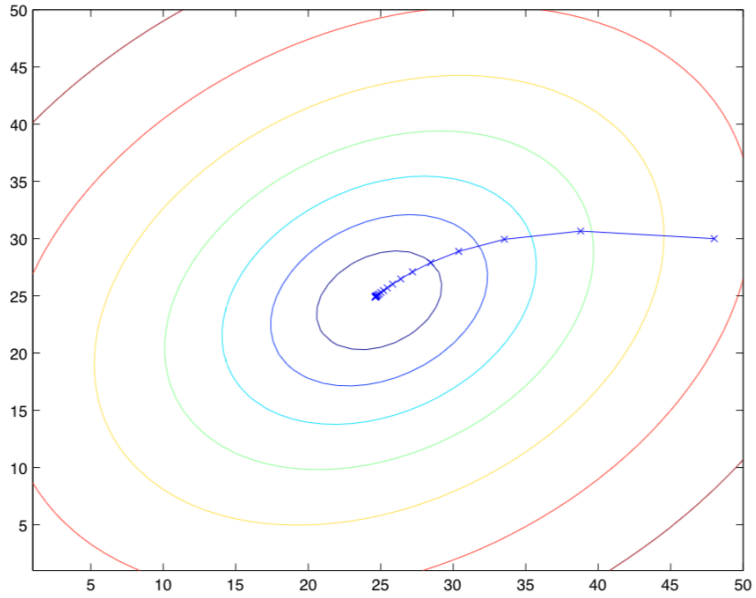
  Hence, if $\nabla f(\boldsymbol{x}_k)_2 \neq 0$ (the first-order necessary condition is not met) and α_k is sufficiently small, we have $f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k)$.

- $\boldsymbol{p}_k$ is the max-rate descend direction: For any $\boldsymbol{p}$ with $\|\boldsymbol{p}\|_2 = 1$, we have

$$\boldsymbol{p}^T \nabla f(\boldsymbol{x}_k) \geq -\|\boldsymbol{p}\|_2 \|\nabla f(\boldsymbol{x}_k)\|_2 = -\|\nabla f(\boldsymbol{x}_k)\|_2.$$

  If we set $\boldsymbol{p} = \boldsymbol{p}_k$, we have $\boldsymbol{p}^T \nabla f(\boldsymbol{x}_k) = -\|\nabla f(\boldsymbol{x}_k)\|_2$.

Steepest descent methods



- The most popular methods (in continuous optimization)
- Simple and intuitive
- Work under very few assumptions (although they cannot directly handle non-differentiable objectives and constraints)
- Suitable for large-scale problems, e.g., easy to parallelize for problems with many terms in the objective
- Usually it's very slow

Search directions: Newton direction

We approximate $f(\boldsymbol{x}_k + \boldsymbol{p})$ by its second-order Taylor series, i.e.

$$f(\boldsymbol{x}_k + \boldsymbol{p}) \approx f(\boldsymbol{x}_k) + \boldsymbol{p}^T \nabla f(\boldsymbol{x}_k) + \frac{1}{2}\boldsymbol{p}^T \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p} \stackrel{\text{def}}{=} m_k(\boldsymbol{p})$$

We minimize the quadratic function m_k instead of the original objective function f. Assume that the Hessian $\nabla^2 f(\boldsymbol{x}_k)$ is positive definite, we obtain the Newton direction as the minimizer of $m_k(\boldsymbol{p})$, i.e.,

$$\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}_k^N = -\nabla f(\boldsymbol{x}_k) \quad \Longrightarrow \quad \boldsymbol{p}_k^N = -\left(\nabla^2 f(\boldsymbol{x}_k)\right)^{-1} \nabla f(\boldsymbol{x}_k).$$

- A new quadratic approximation will be constructed at $\boldsymbol{x}_{k+1}$ again.
- **Special case:** The objective is quadratic. Then, the approximation is exact and the method returns a solution in one step.

Newton's method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^N$$

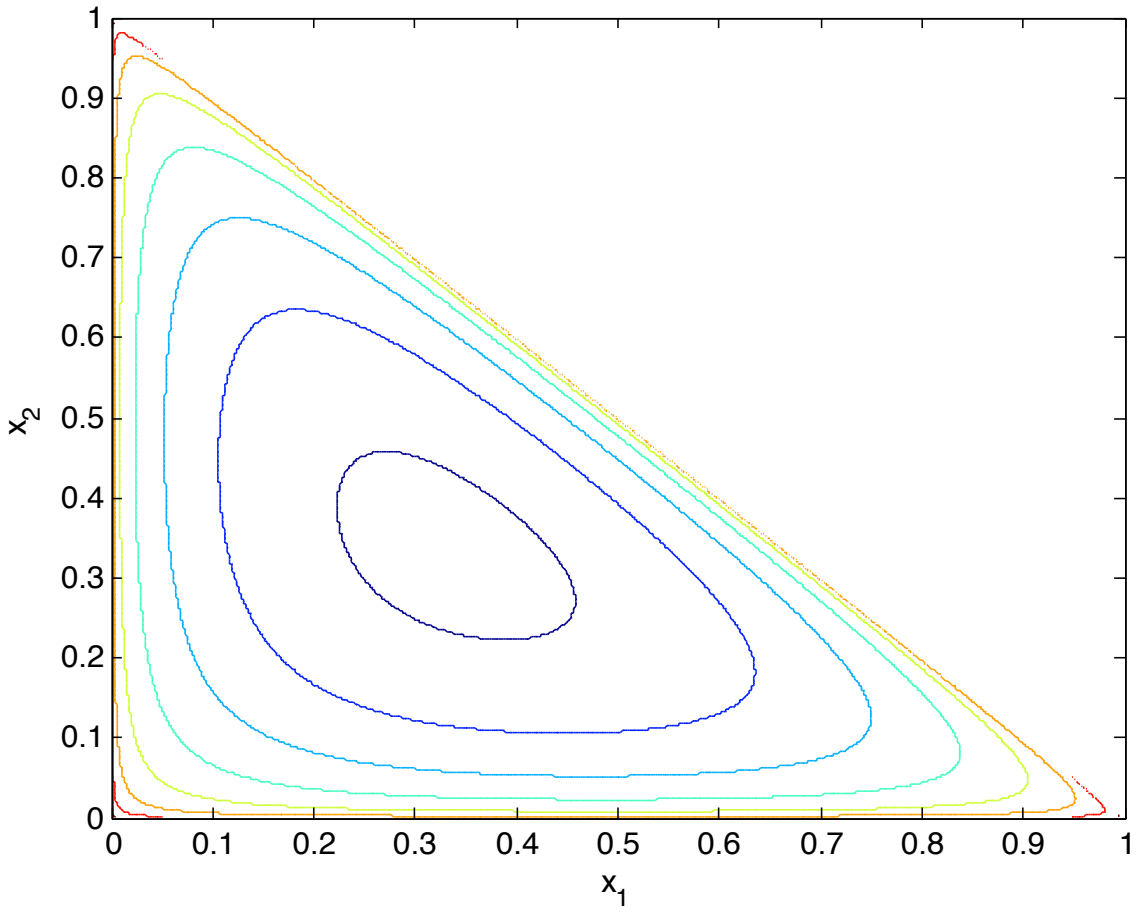
where

$$\boldsymbol{p}_k^N = -\left(\nabla^2 f(\boldsymbol{x}_k)\right)^{-1} \nabla f(\boldsymbol{x}_k).$$

- Need both the gradient and the Hessian
- Based on local quadratic approximations to the objective function
- Requires a positive definite Hessian to work
- Converges very quickly near the solution
- Need a lot of computational work at each iteration:
  - Forming the Hessian
  - Inverting or factorizing the (approximate) Hessian

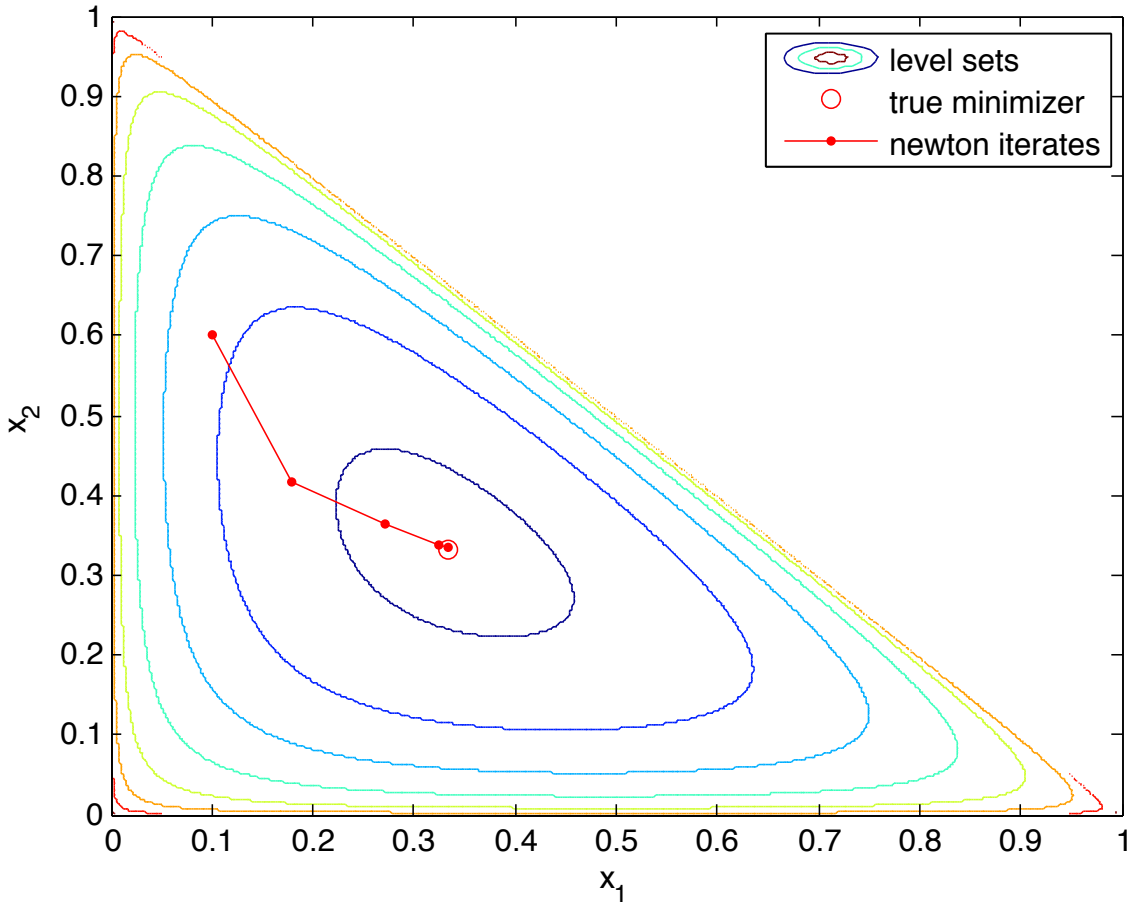
Example: Newton's method

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} f(\boldsymbol{x}) = -\log(1 - x_1 - x_2) - \log(x_1) - \log(x_2)$$



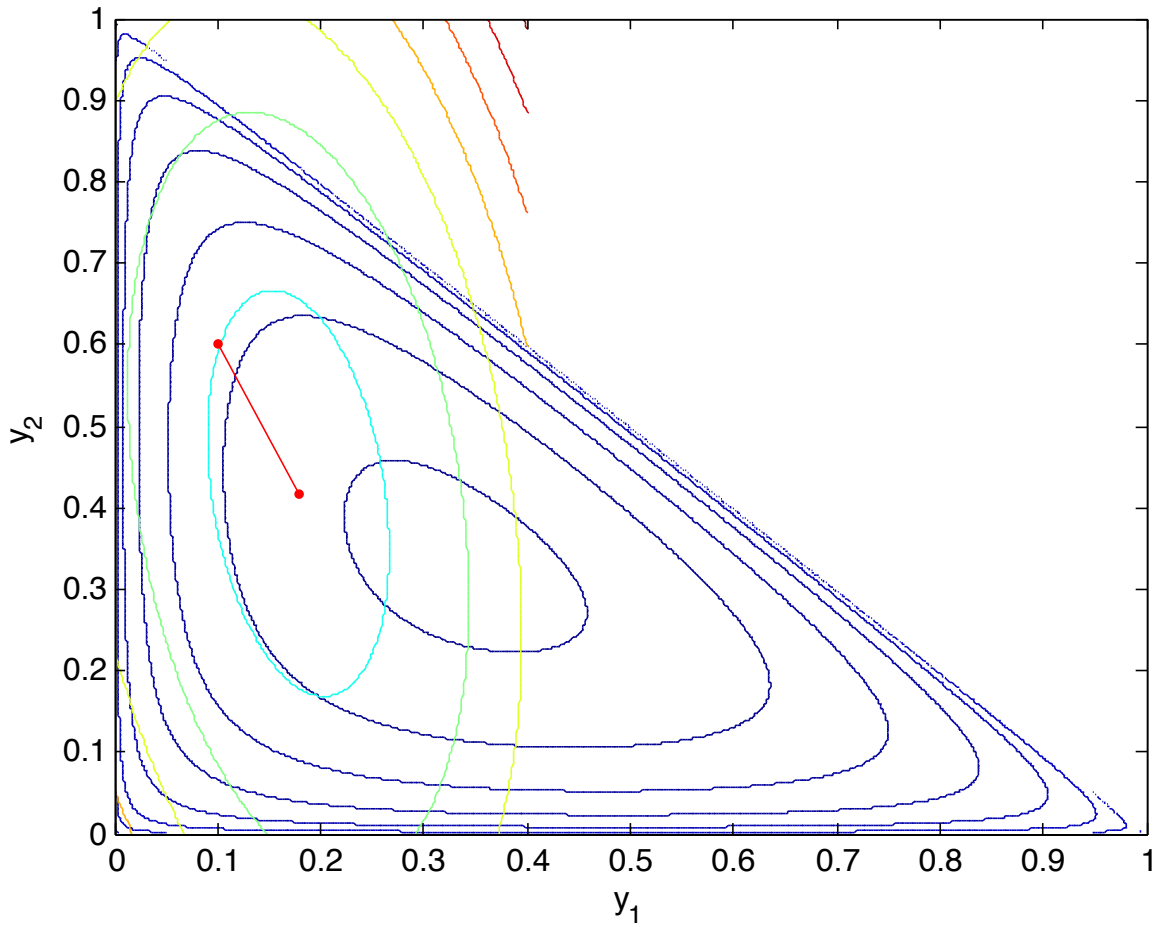
Example: Newton's method

Start the Newton's method from $\boldsymbol{x}_0 = [0.1, 0.6]^T$.



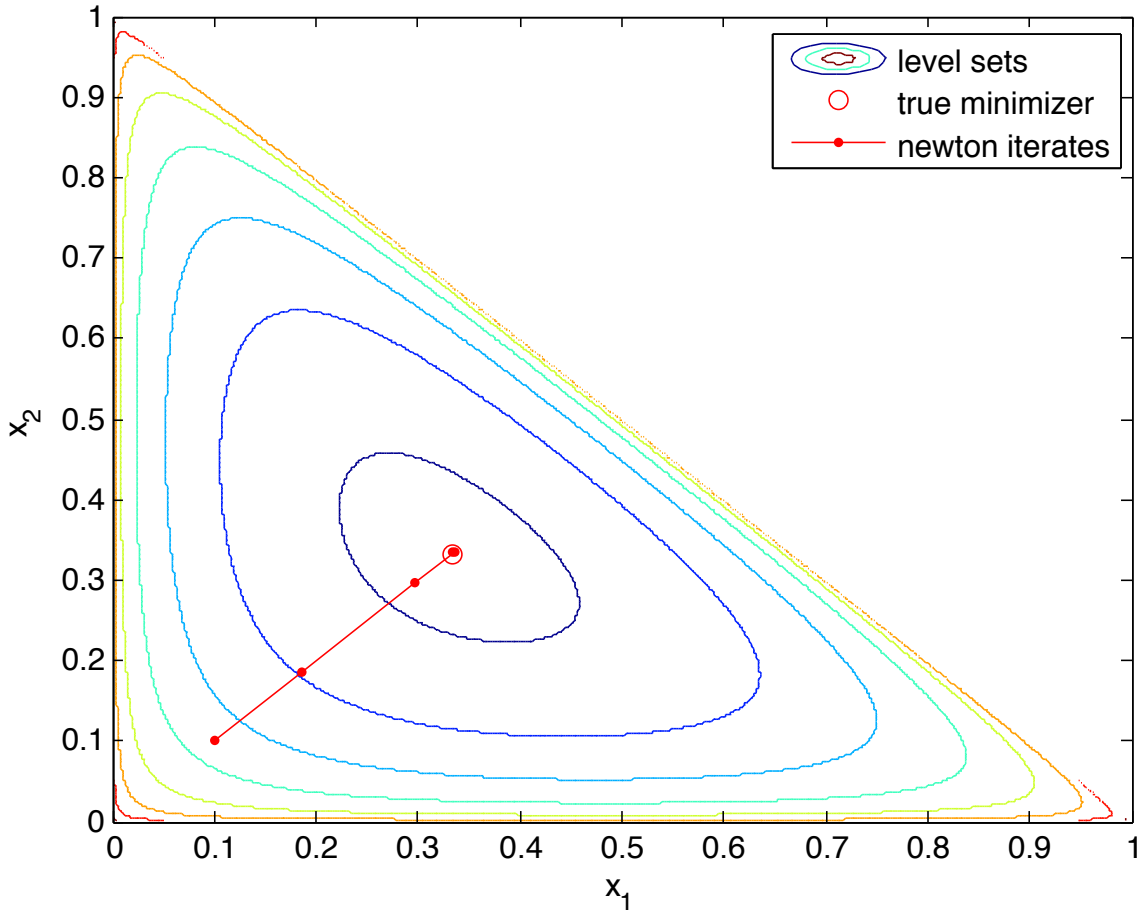
Example: Newton's method

$f(\boldsymbol{x}_0)$ and its quadratic approximation m_0 at $[0.1, 0.6]^T$ share the same value, gradient and Hessian. The new point minimizes m_0.



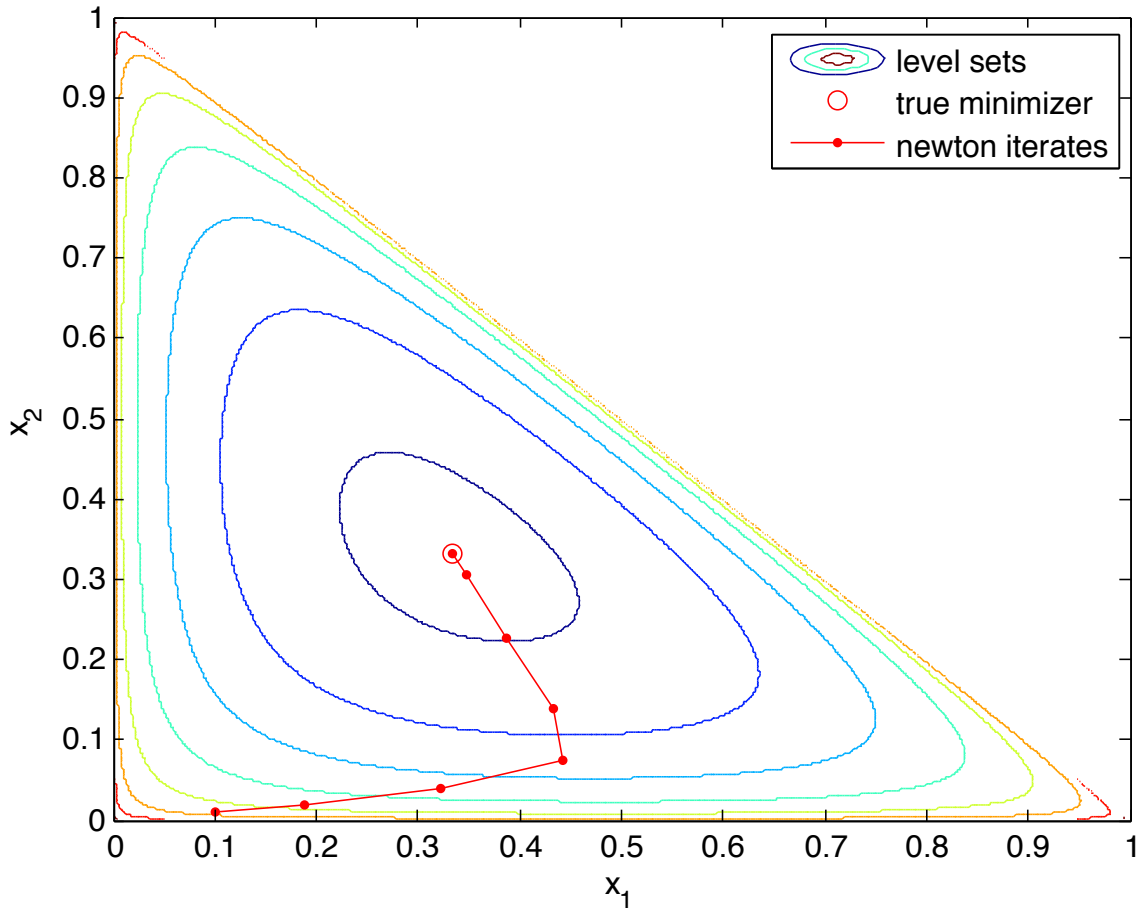
Example: Newton's method

Start the Newton's method from $\boldsymbol{x}_0 = [0.1, 0.1]^T$.



Example: Newton's method

Start the Newton's method from $\boldsymbol{x}_0 = [0.1, 0.01]^T$.



Newton's method: Main issues with Hessian

- **Hessian evaluation:** When the dimension n is large, obtain the Hessian can be computationally expensive.
  - **Solution:** We can use quasi-Newton methods to alleviate this difficulty. (Chapter 6)
- **Indefinite Hessian:** When the Hessian is not positive definite, the direction is not necessarily descending.
  - **Solution:** There are simple modifications. (Chapter 3.4)

Newton's method with Hessian modification

Newton direction $\boldsymbol{p}_k^N$ defined by

$$\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}_k^N = -\nabla f(\boldsymbol{x}_k)$$

Strategy:

- Use $\nabla^2 f(\boldsymbol{x}_k)$, if $\nabla^2 f(\boldsymbol{x}_k) \succ 0$ and its smallest eigenvalue $\lambda_{\min}(\nabla^2 f(\boldsymbol{x}_k)) > \epsilon$; otherwise,
- use $B_k = \nabla^2 f(\boldsymbol{x}_k) + E_k$, so that $B_k \succ 0$ and $\lambda_{\min}(B_k) > \epsilon$.

Newton's method with Hessian modification

- **Method 1: Eigenvalue modification:** Replace negative eigenvalues by
  - a small positive number δ that is somewhat larger than machine precision u, say $\delta = \sqrt{u}$; or
  - its negative values.

  It is computational expensive to find all negative eigenvalues.

- **Method 2: Adding a multiple of the identity:** Replace $\nabla^2 f(\boldsymbol{x}_k)$ by

$$B_k = \nabla^2 f(\boldsymbol{x}_k) + \tau I$$

  with $\tau > 0$ such that $B_k \succ 0$.
  It shifts every eigenvalue of $\nabla^2 f(\boldsymbol{x}_k)$ up by τ.

Newton's method with Hessian modification

- **Method 3: Modified Cholesky Factorization:** Any symmetric matrix $A \succ 0$ can be factored as

$$A = \tilde{L}\tilde{L}^T \qquad \text{or} \qquad A = LDL^T,$$

where $\tilde{L}$ and L are lower triangular matrices, D is a positive diagonal matrix, and L has ones on its main diagonal.

Properties of the Cholesky factorization:

- Very useful in solving linear systems of equations. Reduce a system to two simple systems.
- The factorization is stable if A is positive definite, i.e., small errors in A will not cause large errors in L or D.
- The cost is $n^3/6 + O(n^2)$, roughly half of Gaussian elimination.

Newton's method with Hessian modification

- **Method 3: Modified Cholesky Factorization:**

$$A = LDL^T$$

If A is indefinite but still symmetric, then

- Cholesky factorization may not exist;
- Cholesky factorization can be not stable;
- D has zero or negative element(s) on its diagonal.

Solution:

- **Idea**: Modify A during the factorization such that all elements in D are sufficiently positive.
- **Tools:** Pivoting, i.e., permute the matrix at each step to pull the largest remaining diagonal element to the pivot position.
- **Effect**: Postpone the modification and keep it as small as possible.
- **Modification**: When no acceptable elements remain,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}$$

replay D_2 by a positive definite matrix and complete the factorization.

Newton's method: Main issues with Hessian

- **Hessian evaluation:** When the dimension n is large, obtain the Hessian can be computationally expensive.
  - **Solution:** We can use quasi-Newton methods to alleviate this difficulty. (Chapter 6)
- **Indefinite Hessian:** When the Hessian is not positive definite, the direction is not necessarily descending.
  - **Solution:** There are simple modifications. (Chapter 3.4)

Search directions: Quasi-Newton direction

Consider the second-order Taylor expansion as an approximation of $f(\boldsymbol{x})$, i.e.

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}_k) + (\boldsymbol{x} - \boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^T \nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k)$$

Compute the gradient on $\boldsymbol{x}$, and obtain

$$\nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) \approx \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}_k).$$

Set $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ and $\boldsymbol{y}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)$. We choose a Hessian approximation B_{k+1} or an inverse Hessian approximation H_{k+1} satisfy the **secant equation:**

$$B_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k \quad \text{or} \quad H_{k+1}\boldsymbol{y}_k = \boldsymbol{s}_k.$$

The quasi-Newton direction is defined as

$$\boldsymbol{p}_k = -B_k^{-1}\nabla f(\boldsymbol{x}_k) \quad \text{or} \quad \boldsymbol{p}_k = -H_k \nabla f(\boldsymbol{x}_k).$$

Search directions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \qquad f:\mathbb{R}^n\to\mathbb{R},\ f\in\mathcal{C}^2(\mathbb{R}^n)$$

The iteration step is essentially in the form of

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k,$$

and the search direction $\boldsymbol{p}_k$ typically is from solving a linear system

$$G_k \boldsymbol{p} = -\nabla f(\boldsymbol{x}_k).$$

The difference of the search directions is mainly from how the Hessian is approximated:

$$\begin{aligned} G_k &= I, && \text{Steepest descent} \\ G_k &= \nabla^2 f(\boldsymbol{x}_k), && \text{Newton} \\ G_k &= B_k, && \text{Quasi-Newton} \end{aligned}$$

Step length

The iteration step is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k.$$

After choosing a direction $\boldsymbol{p}_k$, we need find a step length α_k.

- Small step length:
  - **Pros:** Iterations are more likely converge.
  - **Cons:** Need more iterations and thus more computational work.
- Large step length:
  - **Pros:** Better use the descent direction, and may reduce the total iterations.
  - **Cons:** Can cause overshooting and zig-zags. If too large, even can lead to diverged iterations.

Line search

The iteration step is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k.$$

The step length α_k is usually chosen by

- Exact line search: Find the global minimizer of the univariate problem:

$$\min_{\alpha>0} \phi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k).$$

- Inexact line search: Find a step length $\alpha > 0$ that only need satisfies certain conditions.

Exact line search: Convex quadratic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}, \qquad Q \succ 0$$

Then the univariate problem for exact line search is

$$\min_{\alpha>0} \phi(\alpha) = f(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k).$$

According to the optimality condition for convex problems, the minimizer α^* should satisfies

$$\phi'(\alpha^*) = \alpha^*\boldsymbol{p}_k^T Q\boldsymbol{p}_k + \boldsymbol{p}_k^T Q\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{p}_k = 0.$$

We obtain

$$\alpha^* = -\frac{\nabla f^T(\boldsymbol{x}_k)\boldsymbol{p}_k}{\boldsymbol{p}_k^T Q\boldsymbol{p}_k}.$$

Exact line search

In general cases, we need solve a nonlinear univariate problem

$$\min_{\alpha>0} \phi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k),$$

then it is necessary to use an iterative procedure.

- According to the first-order necessary condition, we need find a positive root for the function $\phi'(\alpha)$. Then, we need to know and evaluate the derivative ϕ'.
- There exist also the methods that only need evaluate ϕ, e.g., Golden section search. (Not covered by the course.)

Methods for finding roots (Course 02601, week 2)

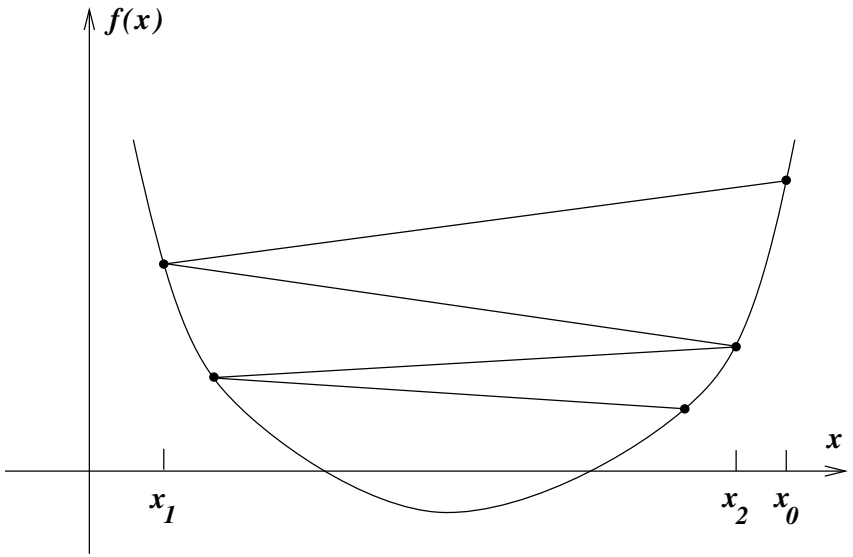
- **Bisection method:**
  - One derivative evaluation at each iteration.
  - Narrow search interval to exactly half each time.
- **Netwon's method:**
  - One derivative and one second derivative evaluations at each iteration.
  - Must start near α^*.
  - Has quadratic convergence.
- **Secant method:**
  - Two points to start with; then one derivative evaluation at each iteration.
  - Must start near α^*.
  - Has superlinear convergence.

Inexact line search

In practice, we usually preform inexact line search to identify a step length that achieves adequate reductions in f at minimal cost.

Idea: Try out a sequence of candidate values for α, stopping to accept one when certain conditions are satisfied.

The intuitive condition $f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) < f(\boldsymbol{x}_k)$ is not enough to produce convergence to $\boldsymbol{x}^*$.

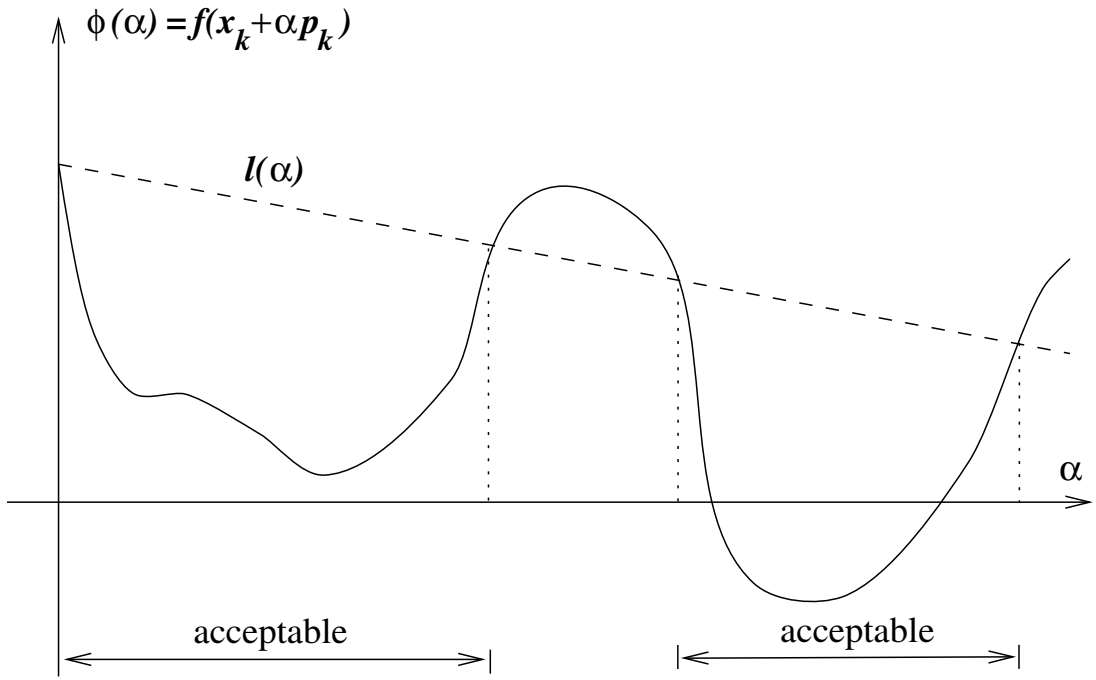


$f(x^*) = -1$, but the descent sequence $\{x_k\}$ gives $f(x_k) = 5/k \to 0$.

Wolfe conditions

- **Armijo condition (sufficient decrease condition):**

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k \quad \text{for some } c_1 \in (0, 1).$$

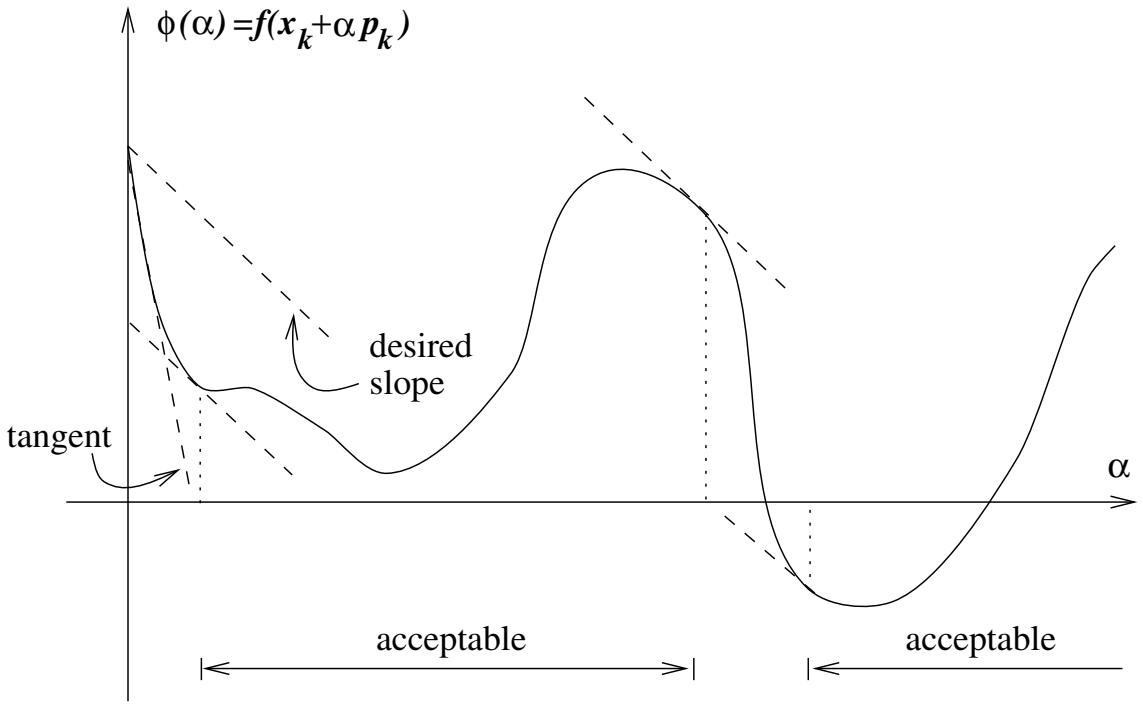


- $l(\alpha) = f(\boldsymbol{x}_k) + c_1 \alpha \nabla f^T(\boldsymbol{x}_k) \boldsymbol{p}_k$
- $l(0) = \phi(0)$
- The slope of $l(\alpha)$, $c_1 \nabla f^T(\boldsymbol{x}_k) \boldsymbol{p}_k$, is negative.

Wolfe conditions

- **Curvature condition:**

$$\nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)^T \boldsymbol{p}_k \geq c_2 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k \quad \text{for some } c_2 \in (c_1, 1).$$



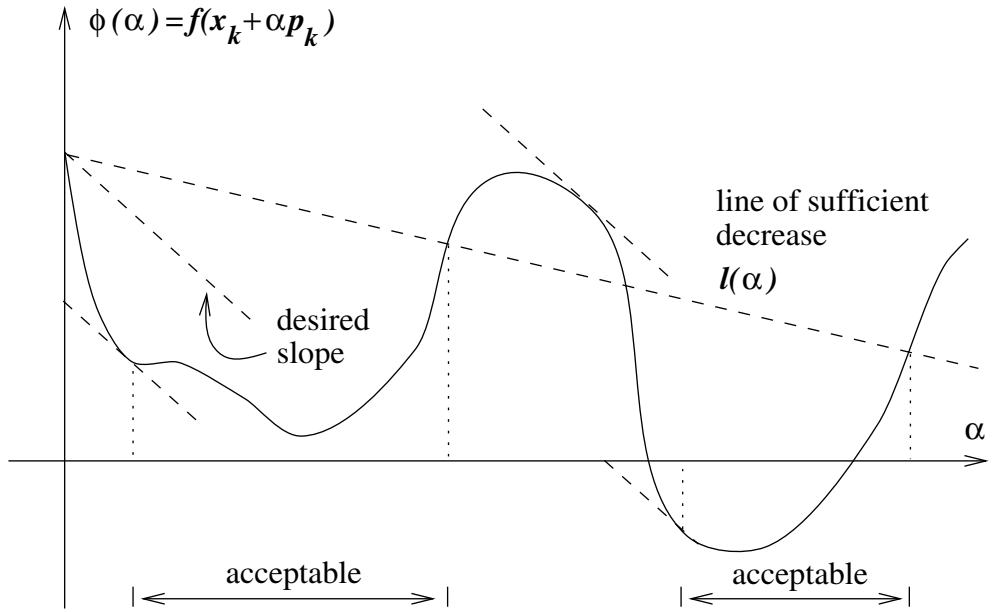
- Ensure the slope of ϕ at α_k is greater than c_2 times the initial slope $\phi'(0)$.
- Avoid very small α.

Wolfe conditions

- **Wolfe conditions:**

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k$$
$$\nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)^T \boldsymbol{p}_k \geq c_2 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k$$

with $0 < c_1 < c_2 < 1$.



- The Wolfe conditions are scale-invariant.

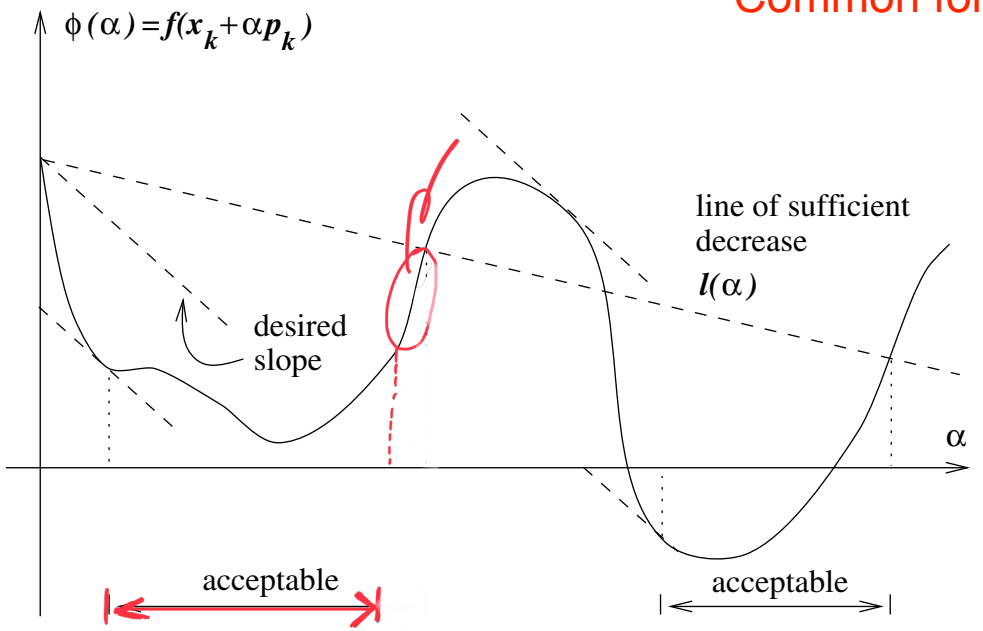
Wolfe conditions

- **Strong Wolfe conditions:**

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c_1 \alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k$$
$$|\nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)^T \boldsymbol{p}_k| \leq c_2 |\nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k|$$

with $0 < c_1 < c_2 < 1$.

Common for quasi-newton

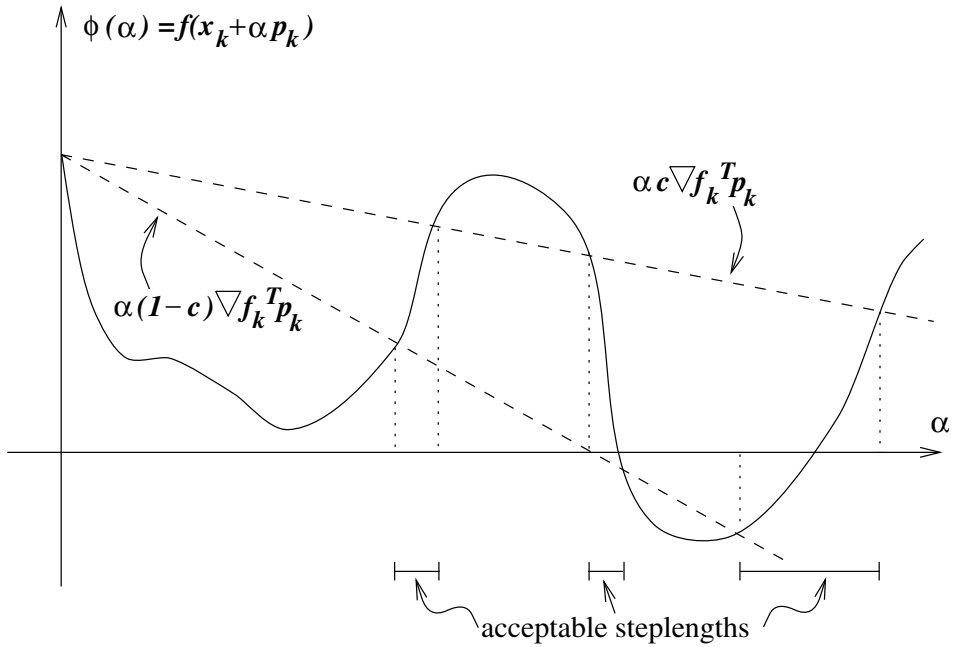


- Avoid the derivative $\phi'(\alpha_k)$ to be too positive.
- For every smooth and bounded below function f, there exist step lengths that satisfy the Wolfe condition and the strong Wolfe condition.

Goldstein conditions

$$f(\boldsymbol{x}_k) + (1-c)\alpha\nabla f(\boldsymbol{x}_k)^T\boldsymbol{p}_k \leq f(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c\alpha\nabla f(\boldsymbol{x}_k)^T\boldsymbol{p}_k$$

with $0 < c < \frac{1}{2}$.



- The first inequality controls the step length from below.
- The second inequality is the Armijo condition.

Backtracking line search

Idea: For a given search direction, start with a relatively large step length, then iteratively shrink the step length (i.e., "backtracking") until a decrease of the objective function is "adequate".

Algorithm

1. Given a descent direction $\boldsymbol{p}_k$ for f. Choose initial $\bar{\alpha} > 0$, $c \in (0, 1)$ and $\rho \in (0, 1)$.
2. Start with $\alpha := \bar{\alpha}$
3. While $f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) > f(\boldsymbol{x}_k) + c\alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k$, set $\alpha := \rho\alpha$.
4. Return $\alpha_k := \alpha$

Interpolation line search

Recall $\phi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)$, then the Armijo condition can be written as

$$\phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0).$$

Idea: For a given search direction, start with a relatively large step length, then iteratively shrink the step length by using the minimizer of a proper approximation of $\phi(\alpha)$ until a decrease of the objective function is "adequate".

Algorithm

1. Given a descent direction $\boldsymbol{p}_k$ for f. Choose initial α_0, $c_1 \in (0, 1)$. Set $i = 0$.
2. While $\phi(\alpha_i) > \phi(0) + c_1 \alpha_i \phi'(0)^T$,
   1. If $i = 0$, we form a quadratic approximation to ϕ by interpolating $\phi(0)$, $\phi'(0)$ and $\phi(\alpha_0)$; otherwise, we form a cubic approximation to ϕ by interpolating $\phi(0)$, $\phi'(0)$, $\phi(\alpha_{i-1})$ and $\phi(\alpha_i)$.
   2. Find the minimizer of the approximation to ϕ and set as α_{i+1}.
   3. Set $i := i + 1$.
3. Return $\alpha_k := \alpha_i$.

Convergence of line search methods

Theorem

Consider any iteration of the form $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$, where $\boldsymbol{p}_k$ is a descent direction and α_k satisfies the Wolfe conditions. Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} \stackrel{\text{def}}{=} \{\boldsymbol{x} : f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$, where $\boldsymbol{x}_0$ is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on $\mathcal{N}$. Then,

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f(\boldsymbol{x}_k)\|_2^2 < \infty.$$

Conditions of the convergence theorem

- The angle θ_k denotes the angle between $\boldsymbol{p}_k$ and the steepest descent direction $-\nabla f(\boldsymbol{x}_k)$, i.e,

$$\cos\theta_k = -\frac{\nabla f(\boldsymbol{x}_k)^T \boldsymbol{p}_k}{\|\nabla f(\boldsymbol{x}_k)\|_2 \|\boldsymbol{p}_k\|_2}.$$

- **Lipschitz continuous gradient:** ∇f is Lipschitz continuous with constant L, if

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2$$

  for all $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{N}$.
- The conditions in the theorem are not too restrictive.

Results of the convergence theorem

- The results hold also for the Goldstein conditions and strong Wolfe conditions.
- The result implies that

$$\cos^2 \theta_k \|\nabla f(\boldsymbol{x}_k)\|_2^2 \to 0.$$

  Then, if the angle θ_k is bounded away from 90°, then we have

$$\lim_{k \to \infty} \|\nabla f(\boldsymbol{x}_k)\|_2 = 0.$$

- The theorem does not guarantee the convergence to a minimizer, but only a stationary point.

Rates of convergence

Let $\{\boldsymbol{x}_k\}$ be a sequence in $\mathbb{R}^n$ that converges to $\boldsymbol{x}^*$. We say that the convergence is

- **Q-linear:** if there is a constant $r \in (0, 1)$ such that

$$\frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|} \leq r \qquad \text{for all } k \text{ sufficiently large.}$$

- **Q-superlinear:** if

$$\lim_{k\to\infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|} = 0.$$

- **Q-quadratic:** if there is a constant $M > 0$ such that

$$\frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2} \leq M \qquad \text{for all } k \text{ sufficiently large.}$$

Convergence rate of steepest descent: Convex quadratic problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}, \qquad Q \succ 0 \tag{1}$$

The unique minimizer $\boldsymbol{x}^*$ must satisfy $Q\boldsymbol{x} = \boldsymbol{b}$.

- The search direction is $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) = \boldsymbol{b} - Q\boldsymbol{x}_k$.
- The exact line search gives us the step length $\alpha_k = -\frac{\nabla f^T(\boldsymbol{x}_k)\boldsymbol{p}_k}{\boldsymbol{p}_k^T Q\boldsymbol{p}_k}$.
- The steepest descent iteration is given by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k$.

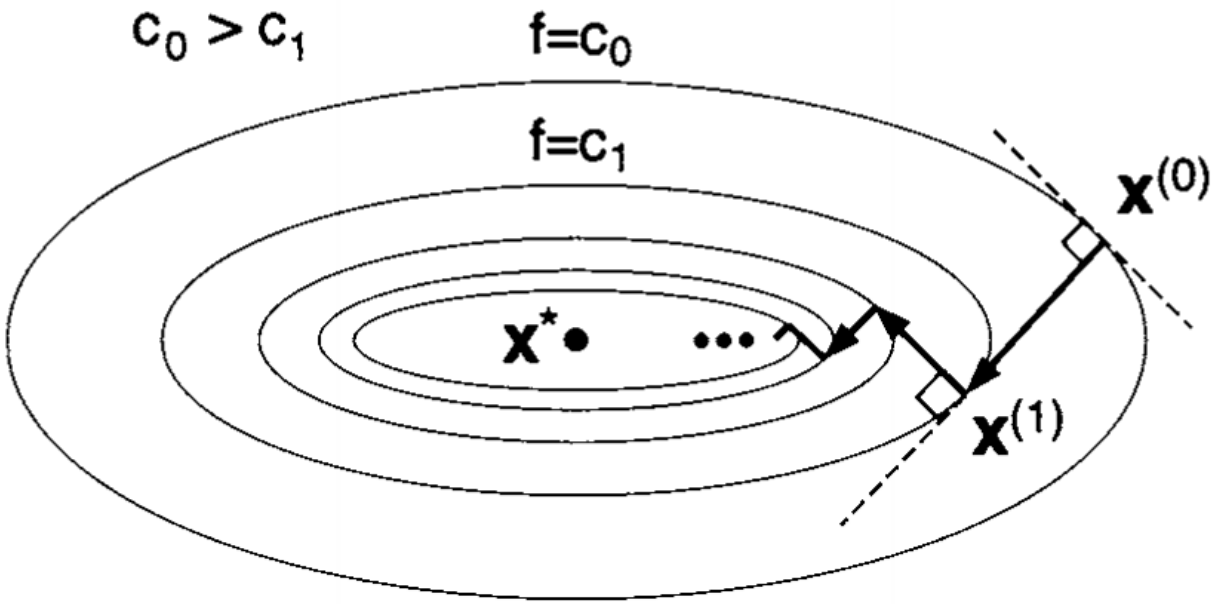
Theorem: When the steepest descent method with exact line searches is applied to (1), the error norm satisfies

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_Q^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_Q^2,$$

where $\|\boldsymbol{x}\|_Q^2 = \boldsymbol{x}^T Q\boldsymbol{x}$, and $0 < \lambda_1 \leq \ldots \leq \lambda_n$ are the eigenvalues of Q.

Examples

- **Example 1:** $f(\boldsymbol{x}) = x_1^2 + x_2^2$. Steepest descent arrives at $\boldsymbol{x}^* = \mathbf{0}$ in 1 iteration.
- **Example 2:** $f(\boldsymbol{x}) = \frac{1}{5}x_1^2 + x_2^2$. Steepest descent makes progress in a narrow valley



Convergence rate of steepest descent

Theorem

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, and that the iterates generated by the steepest descent method with exact line searches converge to a point $\boldsymbol{x}^*$ at which the Hessian matrix $\nabla^2 f(\boldsymbol{x}^*)$ is positive definite. Let r be any scalar satisfying

$$r \in \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1 \right),$$

where $\lambda_1 \leq \ldots \leq \lambda_n$ are the eigenvalues of $\nabla^2 f(\boldsymbol{x}^*)$. Then for all k sufficiently large, we have

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^*) \leq r^2 (f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)).$$

- The theorem shows the linear convergence rate of the steepest descent methods.

Convergence rate of Newton's method

Theorem

Suppose that f is twice differentiable and that the Hessian $\nabla^2 f(\boldsymbol{x})$ is Lipschitz continuous in a neighborhood of a solution $\boldsymbol{x}^*$ at which the sufficient conditions are satisfied. Consider the iteration $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$, where $\boldsymbol{p}_k = -\nabla^2 f(\boldsymbol{x}_k)^{-1} \nabla f(\boldsymbol{x}_k)$. Then,

1. if the starting point $\boldsymbol{x}_0$ is sufficiently close to $\boldsymbol{x}^*$, the sequence of iterates converges to $\boldsymbol{x}^*$;
2. the rate of convergence of $\{\boldsymbol{x}_k\}$ is quadratic; and
3. the sequence of gradient norms $\|\nabla f(\boldsymbol{x}_k)\|_2$ converges quadratically to zero.

Convergence rate of quasi-Newton method

Theorem

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Consider the iteration $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$ and that $\boldsymbol{p}_k = -B_k^{-1}\nabla f(\boldsymbol{x}_k)$. Let us assume also that $\{\boldsymbol{x}_k\}$ converges to a point $\boldsymbol{x}^*$ such that $\nabla f(\boldsymbol{x}^*) = 0$ and $\nabla^2 f(\boldsymbol{x}^*)$ is positive definite. Then $\{\boldsymbol{x}_k\}$ converges superlinearly if and only if

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(\boldsymbol{x}^*))\boldsymbol{p}_k\|_2}{\|\boldsymbol{p}_k\|_2} = 0.$$

- The sequence of quasi-Newton matrices B_k is not necessary to converge to $\nabla^2 f(\boldsymbol{x}^*)$. It suffices that the B_k become increasingly accurate approximations to $\nabla^2 f(\boldsymbol{x}^*)$ along the search directions $\boldsymbol{p}_k$.