

AI505
Optimization

Local Descent

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Line Search Methods
Convergence Analysis
Trust Region Methods

1. Line Search Methods
2. Convergence Analysis
3. Trust Region Methods

For multivariate functions, we have argued that:

- derivatives can have exponential growth in the resulting analytical expression
- calculating zeros might be challenging

Hence, minimizing by solving $\nabla f(\mathbf{x}) = 0$ may be computationally demanding.

Outline

Line Search Methods
Convergence Analysis
Trust Region Methods

1. Line Search Methods
2. Convergence Analysis
3. Trust Region Methods

Descent Direction Iteration

Descent Direction Methods use a local model to incrementally improve design point until some convergence criteria is met.

1. Check termination conditions at \mathbf{x}_k ; if not met, continue.
2. Decide **descent direction** \mathbf{d}_k using local information, commonly required that $\mathbf{d}_k^T \nabla f(\mathbf{x}_k) < 0$.
3. Decide **step size** (ie, magnitude of the overall step that depends on α_k , sometimes but not always $\|\mathbf{d}_k\|_2 = 1$)
4. Compute next design point \mathbf{x}_{k+1}

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

Descent Direction

The search direction often has the form

$$\mathbf{d}_k = -B_k^{-1} \nabla f(\mathbf{x}_k)$$

where B_k is a symmetric and nonsingular matrix.

- in the steepest descent method, B_k is the identity matrix I
- in Newton's method, B_k is the exact Hessian $\nabla^2 f(\mathbf{x}_k)$.
- in quasi-Newton methods, B_k is an approximation to the Hessian that is updated at every iteration by means of a low-rank formula.

When $\mathbf{d}_k = -B_k^{-1} \nabla f(\mathbf{x}_k)$ and B_k is positive definite, we have

$$\mathbf{d}_k^T \nabla f(\mathbf{x}_k) = -\nabla f(\mathbf{x}_k)^T B_k^{-1} \nabla f(\mathbf{x}_k) < 0$$

and therefore \mathbf{d}_k is a descent direction.

- we discuss how to choose α_k and d_k to promote convergence from remote starting points.
- We also consider the rate of convergence of steepest descent, quasi-Newton, and Newton methods.

Line Search for Step Size

Assuming we have the search direction:

- Use it to compute α
- Using the techniques discussed in the previous class, solve:

$$\text{minimize}_{\alpha} f(\mathbf{x} + \alpha \mathbf{d})$$

```
def line_search(f, x, d)
    objective = lambda alpha: f(x + alpha * d)
    a, b = bracket_minimum(objective)
    alpha = minimize(objective, a, b)
    return x + alpha * d
```

Often computationally costly, so approximate line search is used instead

Line Search: Alternatives

α is called to the **learning rate** or **step factor**:

Equal to the $\|\textit{stepsize}\|$ only when $\|\mathbf{d}_k\|_2 = 1$.

- Fixed learning rate
- **Decaying step factor**

$$\alpha_k = \alpha_1 \gamma^{k-1} \quad \text{for } \gamma \in [0, 1]$$

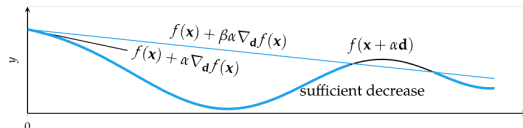
Decaying step factor is often required in convergence proofs

Approximate Line Search

- If function calls are expensive, rather than finding the minimum along a search direction, find a point of sufficient decrease

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \beta \alpha \nabla_{\mathbf{d}_k} f(\mathbf{x}_k)$$

- $\beta \in [0, 1]$, usually $\beta = 1 \times 10^{-4}$

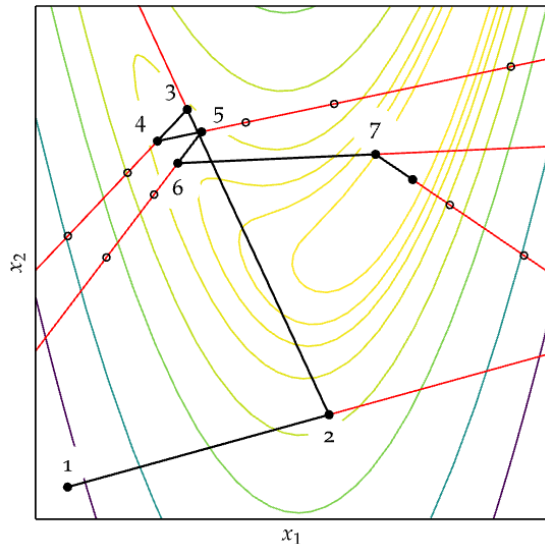


- Alone this condition is insufficient to guarantee convergence to a local $\hat{\alpha}$ minimum. It can converge prematurely.
- Backtracking line search starts with a large step and then backs off

```
def backtracking_line_search(f, grad, x, d, alpha_0=1, p=0.5, beta=1e-4):  
    y, g, alpha = f(x), grad(x), alpha_0  
    while ( f(x + alpha * d) > y + beta * alpha * np.dot(g, d) ) :  
        alpha *= p  
    return alpha
```

- Guaranteed to converge to a local minimum, but can be slow.

Approximate Line Search: Example



Approximate Line Search

Building on backtracking line search are the **Wolfe Conditions** together sufficient to guarantee convergence to a local minimum.

1. First Wolfe Condition: Sufficient Decrease (Armijo condition)

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \beta \alpha \nabla_{\mathbf{d}_k} f(\mathbf{x}_k)$$

2. Second Wolfe Condition: Curvature Condition

$$\nabla_{\mathbf{d}_k} f(\mathbf{x}_{k+1}) \geq \sigma \nabla_{\mathbf{d}_k} f(\mathbf{x}_k)$$

$\beta < \sigma < 1$ with

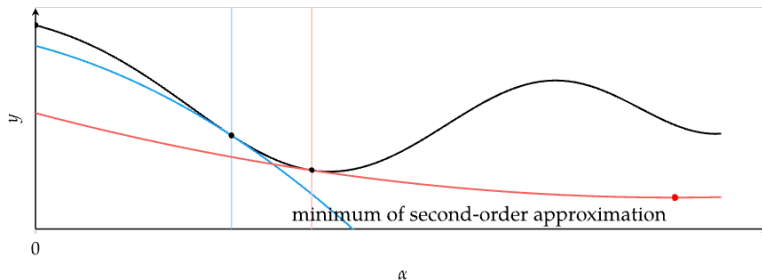
- $\sigma = 0.1$ with conjugate gradient method

- $\sigma = 0.9$ with Newton method

Approximate Line Search

The curvature condition ensures the second-order function approximations have positive curvature

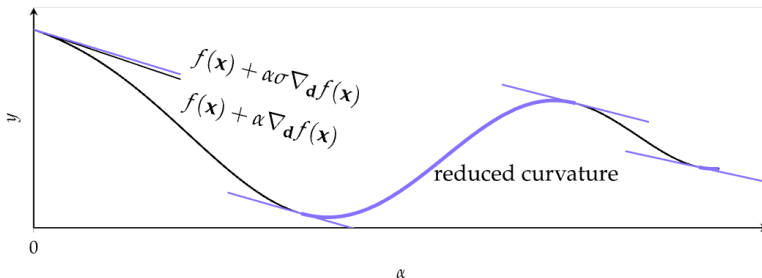
$$\nabla_{d_k} f(\mathbf{x}_{k+1}) \geq \sigma \nabla_{d_k} f(\mathbf{x}_k)$$



Approximate Line Search

Regions satisfying the curvature condition $\nabla_{\mathbf{d}_k} f(\mathbf{x}_{k+1}) \geq \sigma \nabla_{\mathbf{d}_k} f(\mathbf{x}_k)$

- Consider the univariate function $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$.
- The left-hand-side is simply the derivative $\phi'(\alpha_k)$, so the curvature condition ensures that the slope of ϕ at α_k is greater than σ times the initial slope $\phi'(0)$.
- If the slope $\phi'(0)$ is negative, then we are looking for a step size α_k such that the slope of ϕ at that point is still negative but not too negative.
- If $\phi'(\alpha_k)$ is only slightly negative or even positive, it is a sign that we cannot expect much more decrease in f in this direction, so it makes sense to terminate the line search.



Approximate Line Search: Example

Consider approximate line search on $f(x_1, x_2) = x_1^2 + x_1x_2 + x_2^2$
from $\mathbf{x} = [1, 2]$ in the direction $\mathbf{d} = [-1, -1]$, gradient at \mathbf{x} is $\mathbf{g} = [4, 5]$
using a maximum step size of 10, a reduction factor of 0.5,
first Wolfe condition parameter $\beta = 1 \times 10^{-4}$, second Wolfe condition parameter $\sigma = 0.9$.

First Wolfe condition ($f(\mathbf{x} + \alpha\mathbf{d}) \leq f(\mathbf{x}) + \beta\alpha(\mathbf{g}^T \cdot \mathbf{d})$):

$$\alpha = 10 : \quad f([1, 2] + 10 \cdot [-1, -1]) \leq 7 + 1 \times 10^{-4} 10 [4, 5]^T [-1, -1] \implies 217 \not\leq 6.991$$

$$\alpha = 10 \cdot 0.5 = 5 : \quad f([1, 2] + 5 \cdot [-1, -1]) \leq 7 + 1 \times 10^{-4} 5 [4, 5]^T [-1, -1] \implies 37 \not\leq 6.996$$

$$\alpha = 2.5 : \quad f([1, 2] + 2.5 \cdot [-1, -1]) \leq 7 + 1 \times 10^{-4} 2.5 [4, 5]^T [-1, -1] \implies 3.25 \leq 6.998$$

The candidate design point $\mathbf{x}' = \mathbf{x} + \alpha\mathbf{d} = [-1.5, -0.5]$ is checked against the second Wolfe condition $\nabla_{\mathbf{d}} f(\mathbf{x}') \geq \sigma \nabla_{\mathbf{d}} f(\mathbf{x})$:

$$[-3.5, -2.5] \cdot [-1, -1] \geq \sigma [4, 5] \cdot [-1, -1] \implies 6 \geq -8.1$$

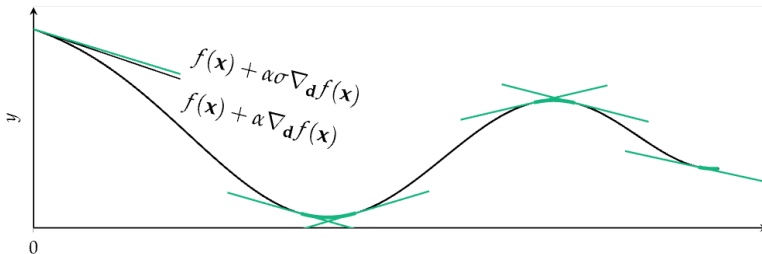
Approximate line search terminates with $\mathbf{x} = [-1.5, -0.5]$.

Approximate Line Search

A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of ϕ . We can modify the curvature condition to force α_k to exclude points that are far from stationary points of ϕ , ie, we no longer allow the derivative $\phi'(\alpha_k)$ to be too positive.

Strong Wolf conditions:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \beta\alpha\nabla_{d_k}f(\mathbf{x}_k) \quad \text{and} \quad |\nabla_{d_k}f(\mathbf{x}_{k+1})| \leq \sigma|\nabla_{d_k}f(\mathbf{x}_k)|$$



Approximate Line Search

- Satisfying the strong Wolfe conditions requires a more complicated algorithm

Strong backtracking line search:

1. Bracketing Phase: tests successively larger step sizes to bracket an interval $[\alpha_{k-1}, \alpha_k]$ guaranteed to contain step lengths satisfying the Wolfe conditions.
2. Zoom Phase: shrink the interval using bisection to find point satisfying the **strong** Wolfe conditions

Approximate Line Search

1. Bracketing Phase

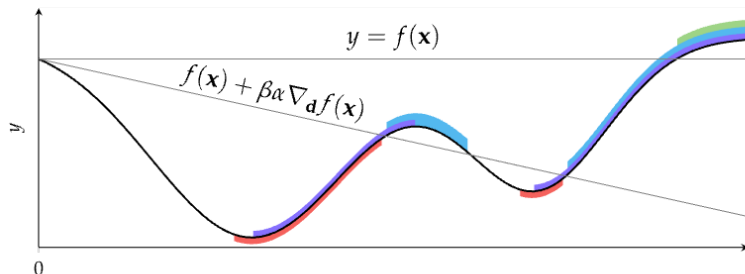
An interval guaranteed to contain step lengths satisfying the Wolfe conditions is found when one of the following conditions hold:

$$f(\mathbf{x} + \alpha \mathbf{d}) \geq f(\mathbf{x})$$

$$f(\mathbf{x} + \alpha \mathbf{d}) > f(\mathbf{x}) + \beta \alpha \nabla \mathbf{d} f(\mathbf{x})$$

$$\nabla f(\mathbf{x} + \alpha \mathbf{d}) \geq 0$$

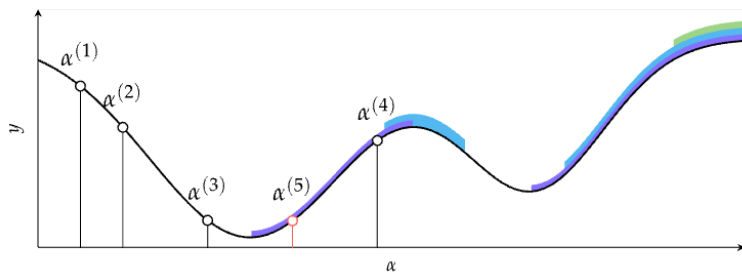
Violation of Wolfe conditions



- $f(\mathbf{x} + \alpha \mathbf{d}) \geq f(\mathbf{x})$
- $f(\mathbf{x} + \alpha \mathbf{d}) > f(\mathbf{x}) + \beta \alpha \nabla_{\mathbf{d}} f(\mathbf{x})$
- $\nabla f(\mathbf{x} + \alpha \mathbf{d}) \geq 0$
- Wolfe conditions satisfied

Approximate Line Search

1. Braketing Phase + zoom phase (α_5)



- $f(\mathbf{x} + \alpha \mathbf{d}) \geq f(\mathbf{x})$
- $f(\mathbf{x} + \alpha \mathbf{d}) > f(\mathbf{x}) + \beta \alpha \nabla_{\mathbf{d}} f(\mathbf{x})$
- $\nabla f(\mathbf{x} + \alpha \mathbf{d}) \geq 0$

Outline

Line Search Methods
Convergence Analysis
Trust Region Methods

1. Line Search Methods
2. Convergence Analysis
3. Trust Region Methods

Convergence Analysis

Line Search Methods
Convergence Analysis
Trust Region Methods

Convergence of Line Search

Let θ_k be the angle between \mathbf{d}_k and the steepest descent direction $-\nabla f_k$, defined by:

$$\cos \theta_k = \frac{-\nabla f_k^T \mathbf{d}_k}{\|\nabla f_k\| \|\mathbf{d}_k\|}$$

Theorem (Zoutendijk condition)

Consider any iteration of the form $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k$, where \mathbf{d}_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 N containing the level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on N , that is, there exists a constant $L > 0$ such that:

$$|f(x) - f(y)| \leq L|x - y|, \quad \forall x, y \in N$$

Then:

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$$

The Zoutendijk condition implies that

$$\cos^2 \theta_k \|\nabla f_k\|^2 \rightarrow 0$$

We can be sure that the gradient norms $\|\nabla f_k\|$ converge to zero, provided that the search directions are never too close to orthogonality with the gradient.

Rate of Convergence

Let $\{\mathbf{x}_k\}$ be a sequence in \mathbb{R}^n that converges to \mathbf{x}^* .

The convergence is said to be **Q-linear** (quotient-linear) if there is a constant $r \in (0, 1)$ such that

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

ie, the distance to the solution \mathbf{x}^* decreases at each iteration by at least a constant factor bounded away from 1 (ie, < 1).

Example:

sequence $\{1 + (0.5)^k\}$ converges Q-linearly to 1, with rate $r = 0.5$.

Rate of Convergence

The convergence is said to be **Q-superlinear** if

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$$

Example: the sequence $\{1 + k^{-k}\}$ converges superlinearly to 1.

An even more rapid convergence rate: The convergence is said to be **Q-quadratic** if

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

where M is a positive constant, not necessarily less than 1. Example: the sequence $\{1 + (0.5)^{2^k}\}$. The values of r and M depend not only on the algorithm but also on the properties of the particular problem. Regardless of these values a quadratically convergent sequence will always eventually converge faster than a linearly convergent sequence.

Rate of Convergence

Superlinear convergence (quadratic, cubic, quartic, etc) is regarded as fast and desirable, while sublinear convergence is usually impractical.

- Steepest descent algorithms converge only at a Q-linear rate, and when the problem is ill-conditioned the convergence constant r in is close to 1.
- Quasi-Newton methods for unconstrained optimization typically converge Q-superlinearly
- Newton's method converges Q-quadratically under appropriate assumptions.

Rate of Convergence

A slightly weaker form of convergence:

overall rate of decrease in the error, rather than the decrease over each individual step of the algorithm.

We say that convergence is **R-linear** (root-linear) if there is a sequence of nonnegative scalars $\{v_k\}$ such that

$$\|\mathbf{x}_k - \mathbf{x}^*\| \leq \{v_k\} \text{ for all } k, \text{ and } \{v_k\} \text{ converges Q-linearly to zero.}$$

Outline

Line Search Methods
Convergence Analysis
Trust Region Methods

1. Line Search Methods
2. Convergence Analysis
3. Trust Region Methods

Trust Region Methods

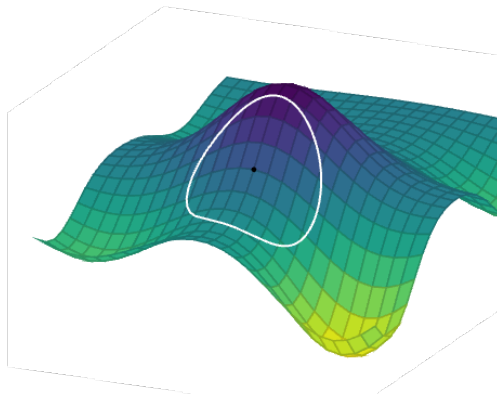
- Descent methods can place too much trust in their first and second order information
- A **trust region** is the local area of the design space where the local model is believed to be reliable.
- Trust region methods, or restricted step methods, limit the step size to ensure local approximation error is minimized
- If the improvement matches the predicted value, the trust region is expanded; otherwise it is contracted

Trust Region Methods

- \mathbf{x}' is new design point
- $\hat{f}(\mathbf{x}')$ is local function approximation, eg, second-order Taylor approximation
- δ is trust region radius

$$\begin{array}{ll}\text{minimize}_{\mathbf{x}'} & \hat{f}(\mathbf{x}') \\ \text{subject to} & \|\mathbf{x} - \mathbf{x}'\| \leq \delta\end{array}$$

Constrained optimization problem.
It can be solved efficiently if \hat{f} quadratic



Trust Region Methods

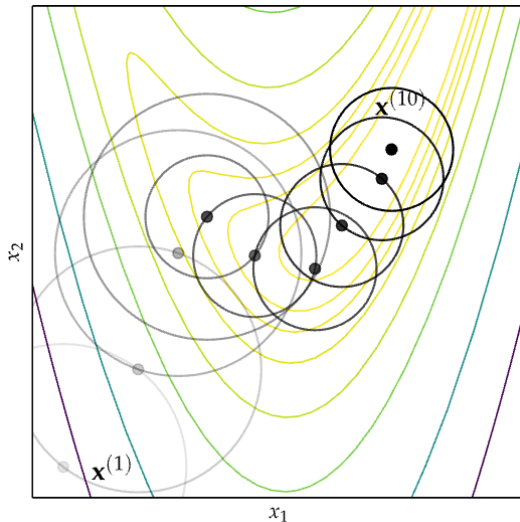
δ can be expanded or contracted based on performance

$$\eta = \frac{\text{actual improvement}}{\text{predicted improvement}} = \frac{f(\mathbf{x}) - f(\mathbf{x}')}{f(\mathbf{x}) - \hat{f}(\mathbf{x}')}$$

If $\eta < \eta_1$ contract by a factor $\gamma_k < 1$

if $\eta > \eta_2$ expand by a factor $\gamma_k > 1$

Trust Region Methods: Example



Trust regions can be also non circular.

Trust Region Methods

Termination Conditions (commonly used together):

- Maximum Iterations: $k > k_{\max}$
- Absolute Improvement: $f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) < \epsilon_a$
- Relative Improvement: $f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) < \epsilon_r |f(\mathbf{x}_k)|$
- Gradient Magnitude: $\|\nabla f(\mathbf{x}_{k+1})\| < \epsilon_g$

Then random restart.

- Descent direction methods incrementally descend toward a local optimum.
- Univariate optimization can be applied during line search.
- Approximate line search can be used to identify appropriate descent step sizes.
- Trust region methods constrain the step to lie within a local region that expands or contracts based on predictive accuracy.
- Termination conditions for descent methods can be based on criteria such as the change in the objective function value or magnitude of the gradient.