Line Search Methods

Numerical Optimization

Prepared by LimCK

Content

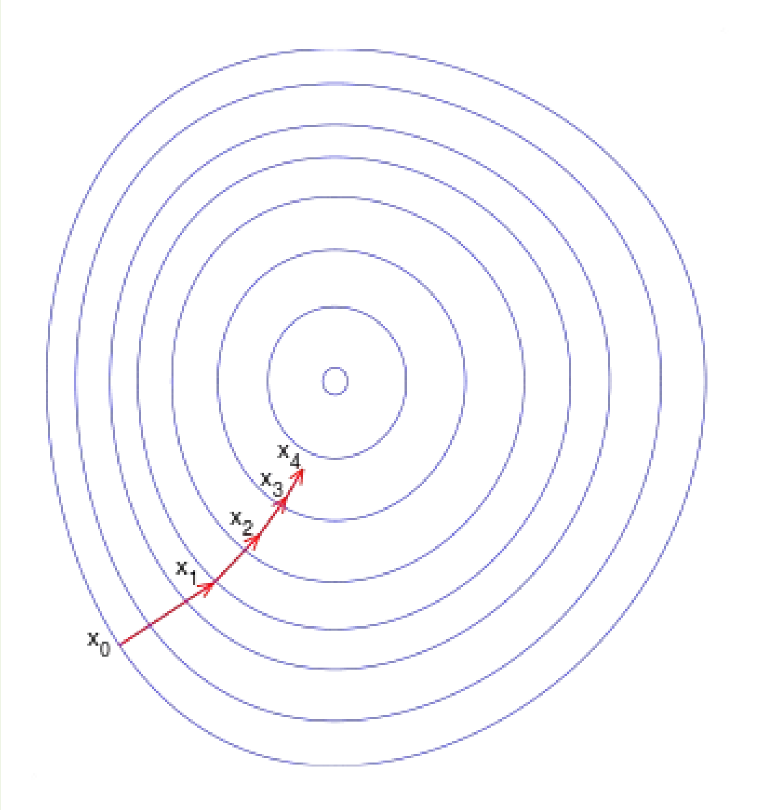
- Line search methods: direction and step length
- Condition of decreases
- Convergence rate

Line Search Methods

- Iteration:

$$x_{k+1} = x_k + \alpha_k p_k \qquad (1)$$

- First, we determine p_k, the search direction at x_k.
- For each search direction, we determine α_k, the step length.
- Move to x_{k+1}.
- Repeat until we reach minimizer.



ө

Search direction, *p*

- Steepest descent method:

$$p_k = -\nabla f_k \tag{2}$$

- Newton method:

$$p_k = -\nabla^2 f_k^{-1} \ \nabla f_k \tag{3}$$

- Quasi Newton method:

$$p_k = -B_k^{-1} \ \nabla f_k \tag{4}$$

Step Length, α

Step Length, α

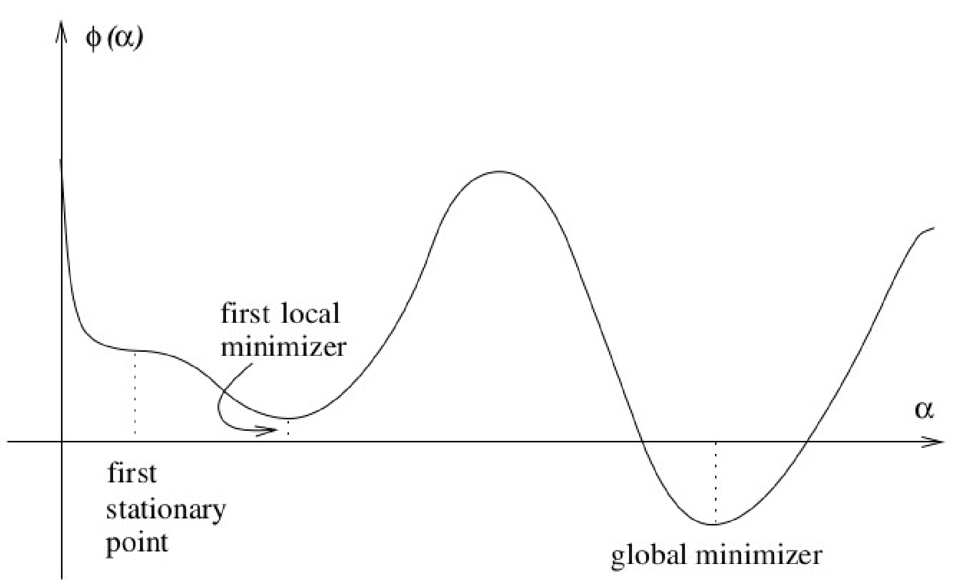
- Tradeoff : we want to get the best α_k, but it is too expensive to compute it.
- So, instead of *exact line search* (compute the best α_k), we use *inexact line search*.
- We find α_k that brings adequate reductions in f at minimal cost.

Step Length, *α*

- Typically: try out a sequence of *α*, stop to accept one of these values when certain conditions are satisfied.
- Assume *Φ* as a function of *α*:

$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0 \tag{5}$$

The graph showing how *f* change with the change of *α*, if x and p are fixed.

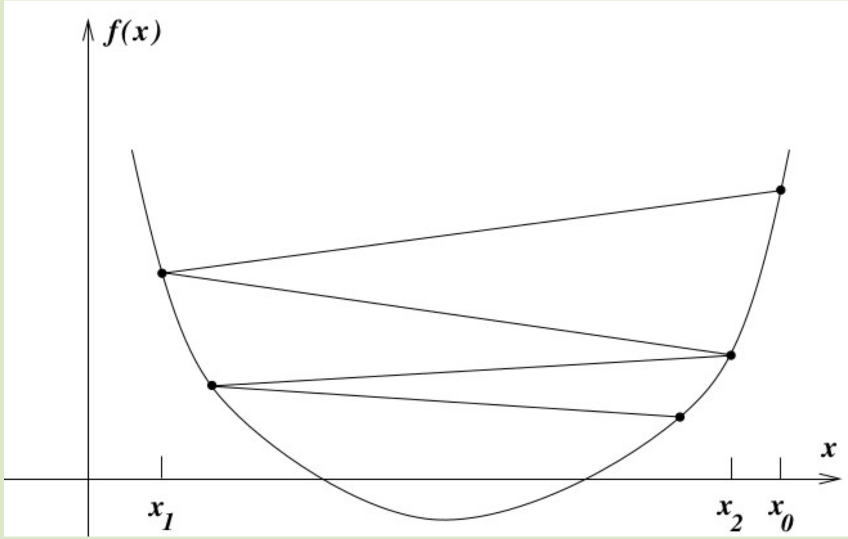


Condition: As long as some reduction?

- As long as a step brings a lower f will do?

$$f(x_k + \alpha_k p_k) < f(x_k)$$

- NO! not enough to produce convergence to *x**
- Example: f* = -1, $f(x_k) = 5/k$, $k = 1, 2, 3, ...$



Sufficient decrease - Armijo condition

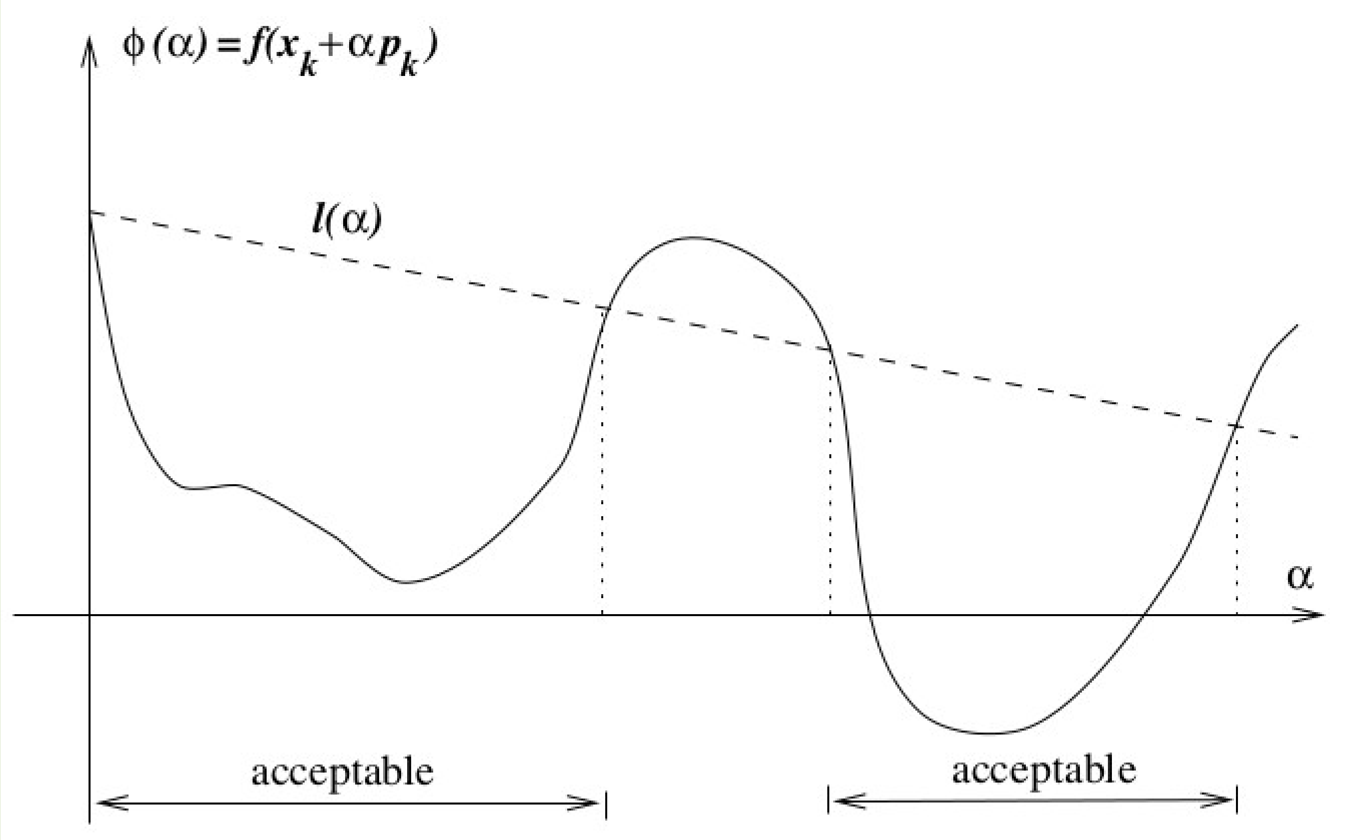
- the reduction in f should be proportional to both the step length α_k and the directional derivative $\nabla f_k^T p_k$.

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k \qquad (6)$$

for some constant $c_1 \in (0, 1)$

- In practice, c_1 normally is a small value, e.g. 10^{-4}
- Observe that the right hand side of (6) is a linear equation $k + m\alpha = l(\alpha)$

Sufficient decrease - Armijo condition



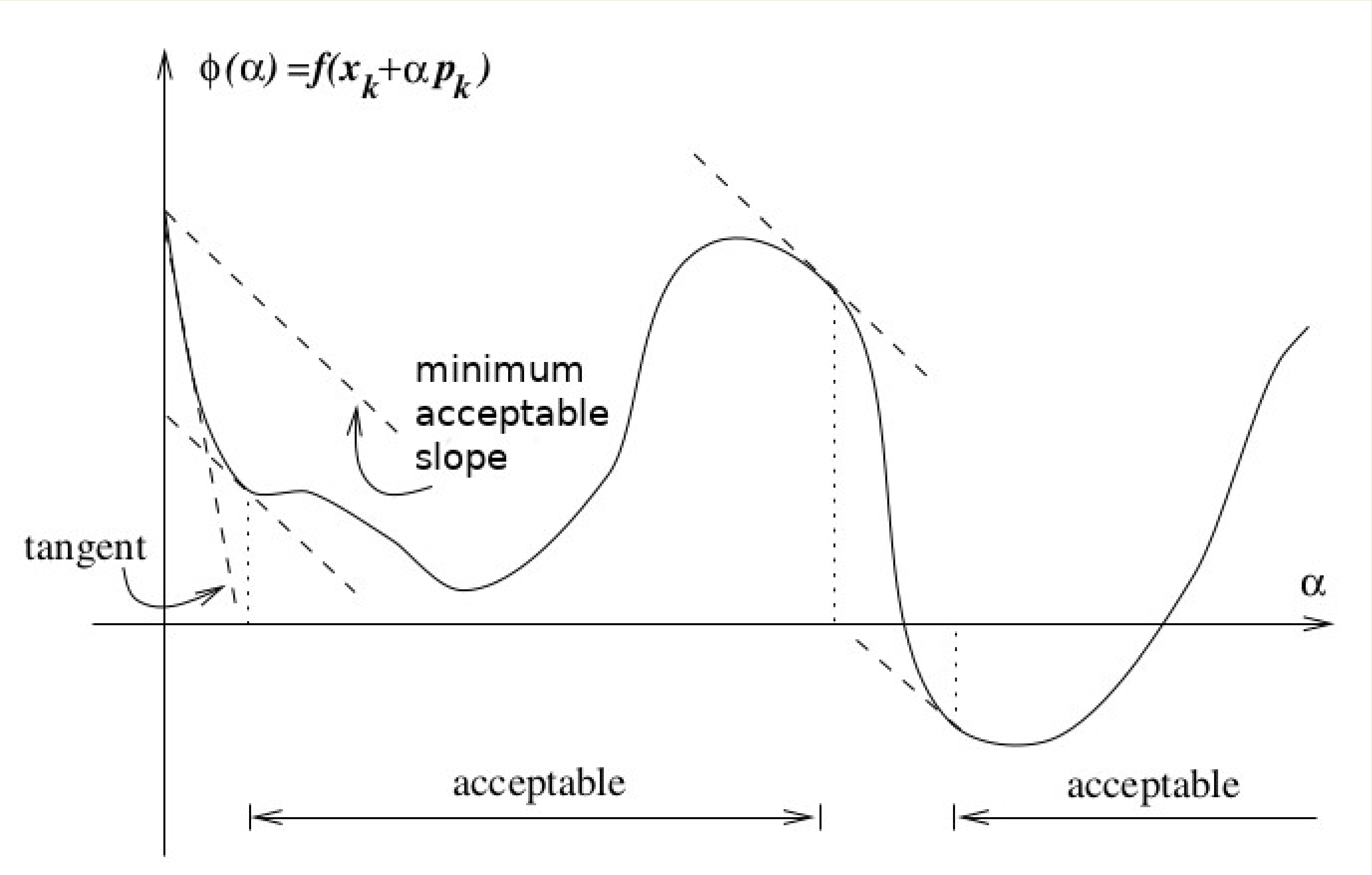
Curvature condition

- To rule out unacceptably short steps we introduce a second requirement – curvature condition:

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k \qquad (7)$$

- c_2 is a constant in (c_1, 1), typically close to 1 (e.g. 0.9)
- This condition means the gradient at x_{k+1} (if the step with α is taken) must be greater than or equal to c_2 times the initial gradient.
- Note: The gradient near the minima should be “less negative” than the gradient at a further point.

Curvature condition



Wolfe Conditions

- The Armijo condition and curvature condition are known collectively as the Wolfe conditions.

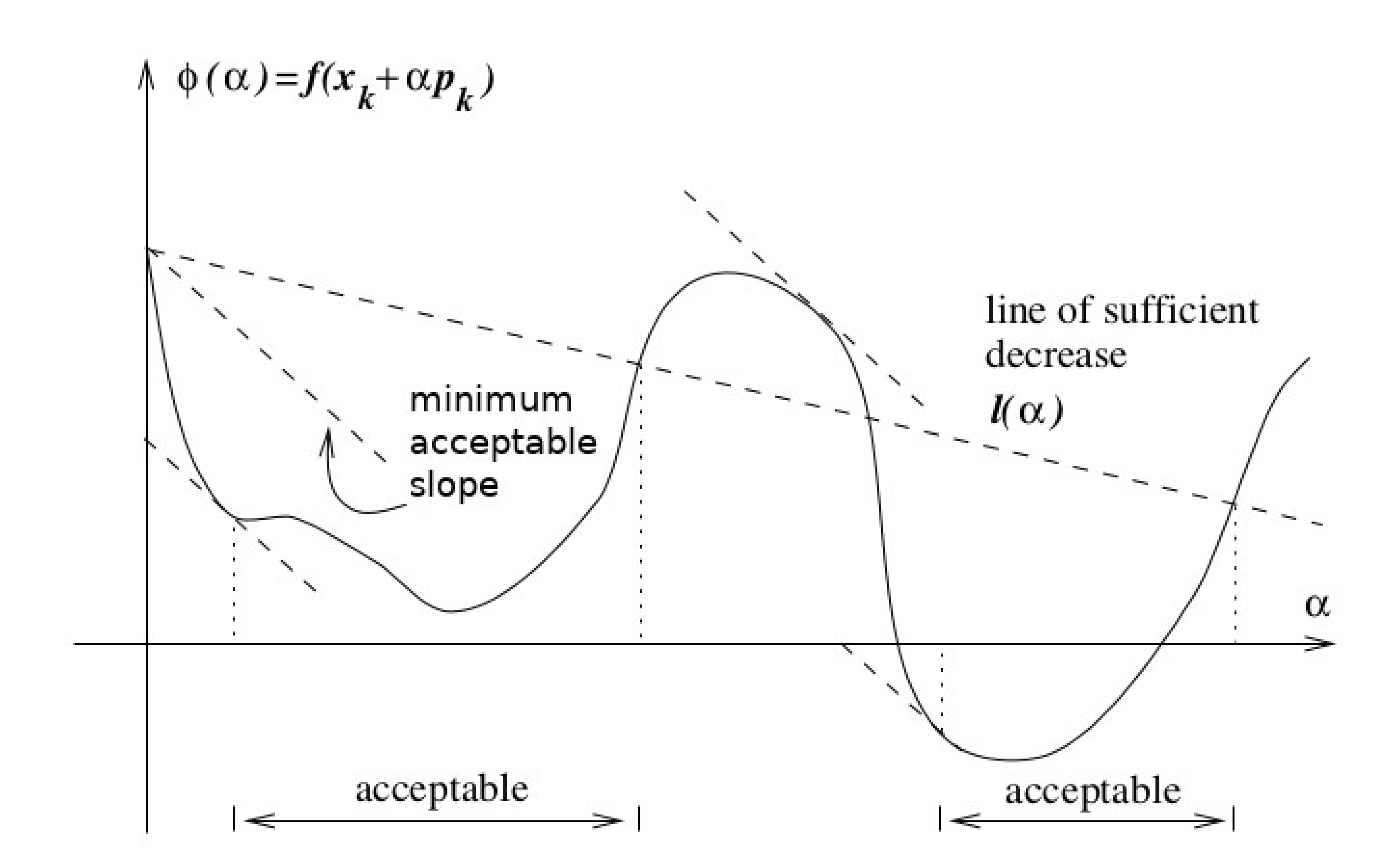
$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$$

with $0 < c_1 < c_2 < 1$

- Can be used in most line search methods

Wolfe Conditions



Goldstein Conditions

- Also ensure that the step length α achieves sufficient decrease but is not too short.
- With $0 < c < 0.5$:

$$f(x_k) + (1-c)\alpha_k \nabla f_k^T p_k \le f(x_k + \alpha_k p_k) \le f(x_k) + c\alpha_k \nabla f_k^T p_k \quad (8)$$

Control step length

Sufficient decrease condition

Backtracking Line Search

- With backtracking, sufficient decrease condition (Armijo) alone is enough.
- This method start from $\alpha > 0$ (e.g., 1), decrease value of α until terminate by sufficient decrease condition.
- Choose the α that just fulfilled sufficient decrease condition.

Backtracking Line Search

Algorithm 3.1 (Backtracking Line Search).

Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;

repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$

$\qquad \alpha \leftarrow \rho\alpha$;

end (repeat)

Terminate with $\alpha_k = \alpha$.

ρ - contraction factor

Convergence of Line Search Methods

- An algorithm is global convergent if:

$$\lim_{k \to \infty} \|\nabla f_k\| = 0$$

- It means, the algorithm convergence to a ~~minimizer~~ stationary point for any starting point x_0.
- Steepest descent method – globally convergent, but may be slow in difficult problems.
- Newton method – not converge if Hessians are not positive definite

Rate of Convergence

Steepest Descent	Newton Method	Quasi Newton Method
• convergence rate is linear • Can be very slow, zigzagging behaviour.	• convergence rate is quadratic if no line search ($\alpha_k = 1$) • Not working if Hessian is not positive definite, or Hessian modification is required.	• convergence rate is superlinear if B_k is getting closer to Hessian along the search. • And also require $\alpha_k = 1$ when it search near the solution.

Rate of Convergence

- If e_n is the error at iteration n, and e_{n+1} is the error at iteration n+1, and:

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^p} = \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = \mu$$

- If $p = 1$, $\mu = 1$: the convergence is sublinear
- If $p = 1$, $0 < \mu < 1$: the convergence is linear. The convergence rate is μ
- If $p = 1$, $\mu = 0$: the convergence is superlinear.
- If $p = 2$, $\mu > 0$: the convergence is quadratic.
- quadratic convergence implies superlinear convergence

Rate of Convergence - example

a =	0.8										
n	1	2	3	4	5	6	7	8	9	10	11
a^n	0.8	0.64	0.512	0.4096	0.32768	0.26214	0.20972	0.16777	0.13422	0.10737	0.0859
a^n / a^{n-1} (linear)		0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
a^d, $d = n^2$	0.8	0.4096	0.13422	0.02815	0.00378	0.00032	1.8E-05	6.3E-07	1.4E-08	2E-10	1.9E-12
a^n / a^{n-1} (superlinear)		0.512	0.32768	0.20972	0.13422	0.0859	0.05498	0.03518	0.02252	0.01441	0.00922
$a^n / (a^{n-1})^2$		0.64	0.8	1.5625	4.76837	22.7374	169.407	1972.15	35873.2	1019579	4.5E+07
a^d, $d = 2^n$	0.64	0.4096	0.16777	0.02815	0.00079	6.3E-07	3.9E-13	1.6E-25	2.4E-50	6E-100	3E-199
a^n / a^{n-1}		0.64	0.4096	0.16777	0.02815	0.00079	6.3E-07	3.9E-13	1.6E-25	2.4E-50	6E-100
$a^n / (a^{n-1})^2$ (quadratic)		1	1	1	1	1	1	1	1	1	1

Exercise

- Given that:

$$f(x) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

- When x_k = [2 1]^T, using SD method, determine the x_{k+1} if α=0.3.
- When x_k = [2 1]^T, using Newton method, determine if the Hessian is positive definite. Find x_{k+1} with α=0.3 if it is positive definite.