Unconstrained Optimization

Unconstrained Optimization

$f : \mathbb{R}^n \to \mathbb{R}:$ $\qquad \min_{x} f(x),$

$x \in \mathbb{R}^n$

Unconstrained Optimization

$f : \mathbb{R}^n \to \mathbb{R}$:

$x \in \mathbb{R}^n$

$$\min_{x} f(x),$$

example

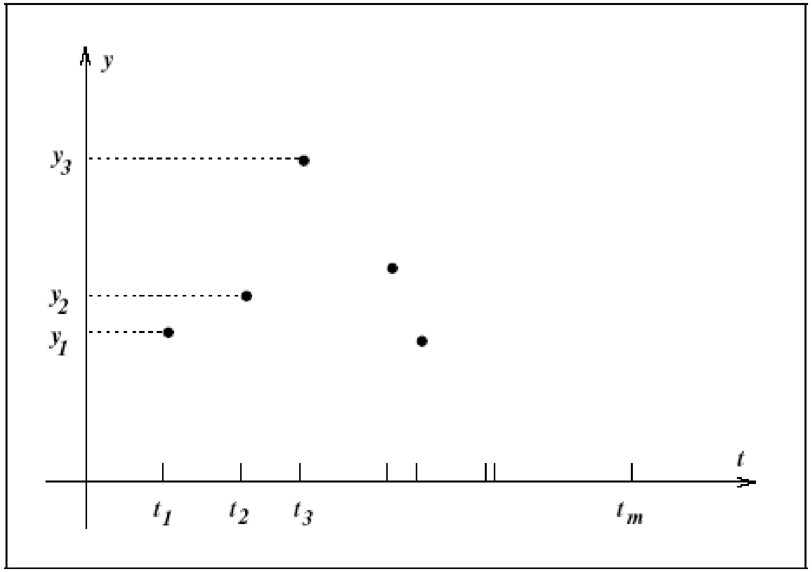
✓find a curve that fits some experimental data

✓Minimize the MSE

$$\phi(t; x) = x_1 + x_2 e^{-(x_3-t)^2/x_4} + x_5 \cos(x_6 t).$$

$$x = (x_1, x_2, \dots, x_6)^T$$

$$r_j(x) = y_j - \phi(t_j; x), \qquad j = 1, 2, \dots, m$$



Unconstrained Optimization

$f : \mathbb{R}^n \rightarrow \mathbb{R}$:
$x \in \mathbb{R}^n$

$$\min_{x} f(x),$$

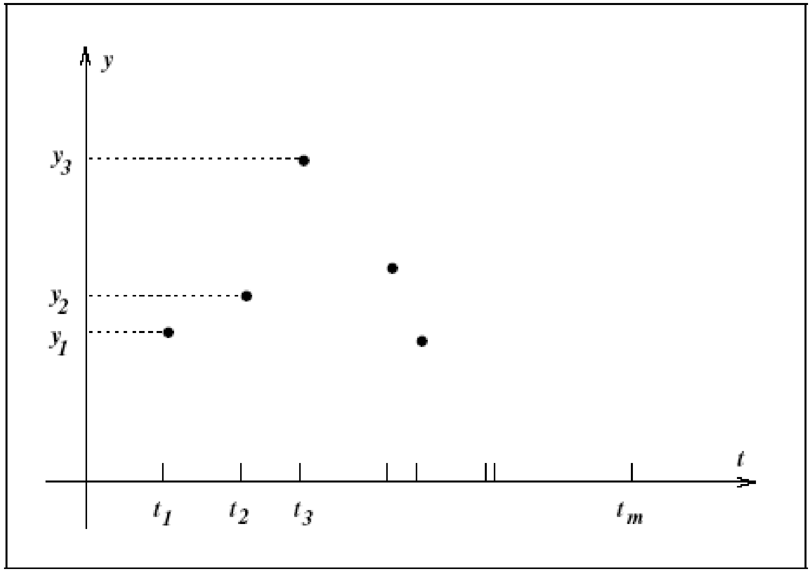
example

✓find a curve that fits some experimental data
✓Minimize the MSE

$$\phi(t;x) = x_1 + x_2 e^{-(x_3-t)^2/x_4} + x_5\cos(x_6 t).$$

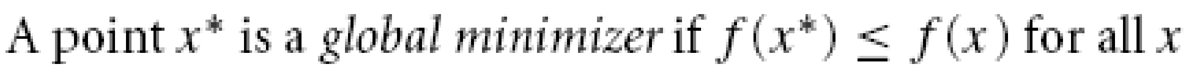
$$x = (x_1, x_2, \ldots, x_6)^T$$

$$r_j(x) = y_j - \phi(t_j; x), \qquad j = 1, 2, \ldots, m$$

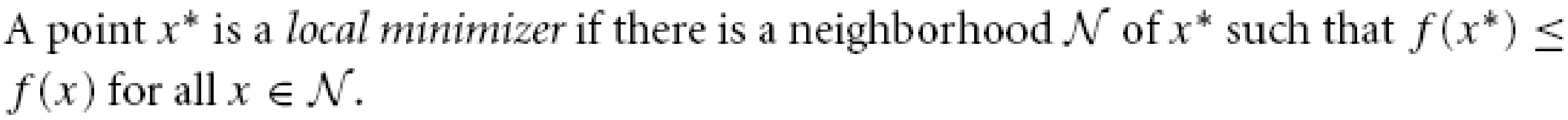


$$\min_{x\in\mathbb{R}^6} f(x) = r_1^2(x) + r_2^2(x) + \cdots + r_m^2(x).$$

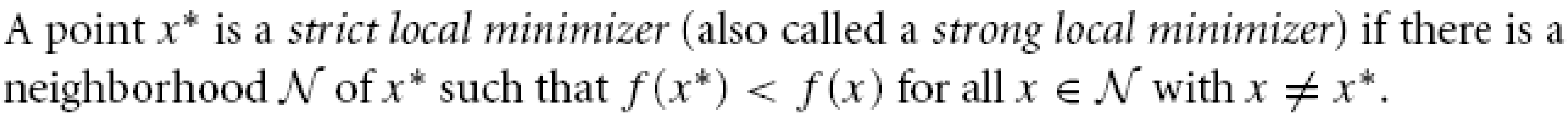
Minimizer



A point x^* is a *global minimizer* if $f(x^*) \leq f(x)$ for all x



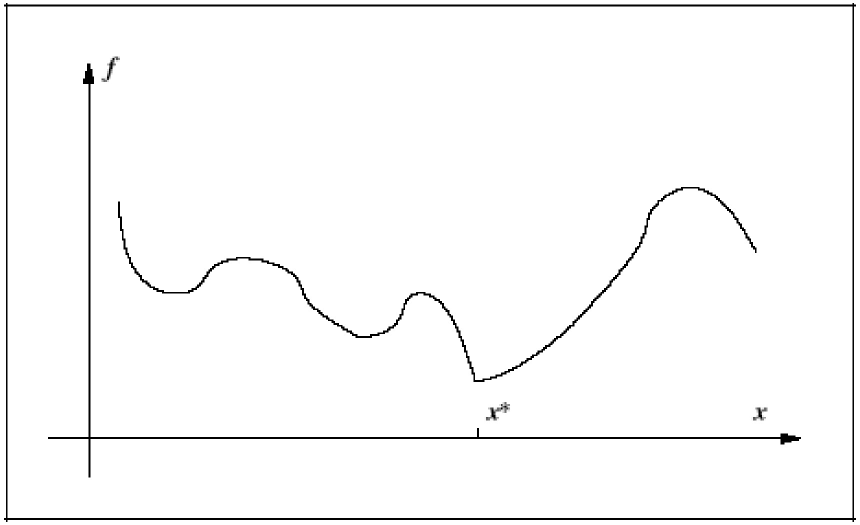
A point x^* is a *local minimizer* if there is a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}$.



A point x^* is a *strict local minimizer* (also called a *strong local minimizer*) if there is a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) < f(x)$ for all $x \in \mathcal{N}$ with $x \neq x^*$.

We focus on smooth functions, functions whose second derivatives exist and are continuous.

Non-smooth Problems



subgradient or generalized gradient

minimizing each smooth piece individually

$$f(x) = \|r(x)\|_1, \qquad f(x) = \|r(x)\|_\infty$$

reformulated as smooth constrained optimization problems

Recognizing a Local Minimum

Theorem (Second-Order Necessary Conditions).
If x^* is a local minimizer of f then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

Theorem (Second-Order Sufficient Conditions).
Suppose that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite.
Then x^* is a strict local minimizer of f.

When f is convex, any local minimizer x^* is a global minimizer of f.

Overview of algorithms

x_0 generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$

terminate : no more progress or a solution point with sufficient accuracy.

two strategies for moving from x_k *to a new* iterate x_{k+1}

Overview of algorithms

x_0 generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$

terminate : no more progress or a solution point with sufficient accuracy.

two strategies for moving from x_k *to a new* iterate x_{k+1}

1. Line search

✓choose a direction p_k *and search along* this direction

$$\min_{\alpha>0} f(x_k + \alpha p_k).$$

Overview of algorithms

2. Trust region

✓construct a model function m_k whose behavior near x_k is similar to f
✓search for a minimizer of *m_k* to some region around *x_k*

$$\min_p m_k(x_k + p), \qquad \text{where } x_k + p \text{ lies inside the trust region.}$$

$$\|p\|_2 \leq \Delta; \qquad m_k(x_k + p) = f_k + p^T \nabla f_k + \tfrac{1}{2} p^T B_k p,$$

Overview of algorithms

2. Trust region

✓construct a model function m_k whose behavior near x_k is similar to f

✓search for a minimizer of m_k to some region around x_k

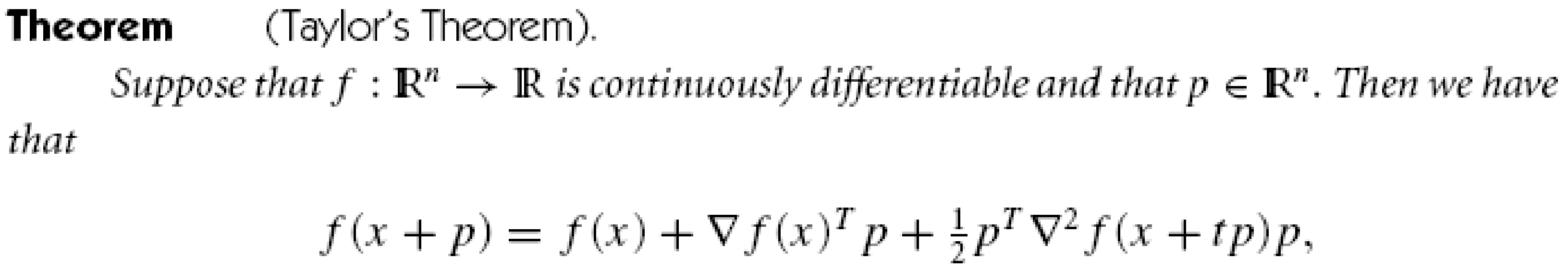
$$\min_p m_k(x_k + p), \qquad \text{where } x_k + p \text{ lies inside the trust region.}$$

$$\|p\|_2 \leq \Delta;$$

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \tfrac{1}{2} p^T B_k p,$$

line search and trust-region approaches differ in the order in which they choose the *direction and distance of the move to the next iterate*

SEARCH DIRECTIONS FOR LINE SEARCH METHODS



Theorem (Taylor's Theorem).

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and that $p \in \mathbb{R}^n$. Then we have that

$$f(x+p) = f(x) + \nabla f(x)^T p + \tfrac{1}{2} p^T \nabla^2 f(x+tp) p,$$

SEARCH DIRECTIONS FOR LINE SEARCH METHODS

Theorem (Taylor's Theorem).

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and that $p \in \mathbb{R}^n$. Then we have that

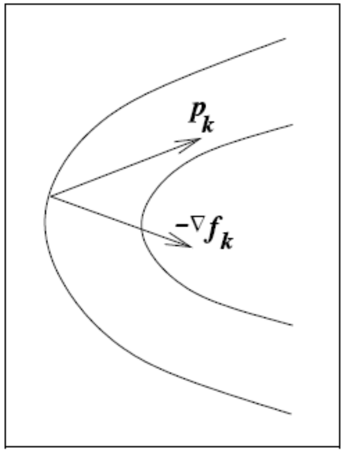
$$f(x+p) = f(x) + \nabla f(x)^T p + \tfrac{1}{2} p^T \nabla^2 f(x+tp)p,$$

$$f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f_k + \tfrac{1}{2}\alpha^2 p^T \nabla^2 f(x_k + tp)p,$$

Descent Methods

Descent methods

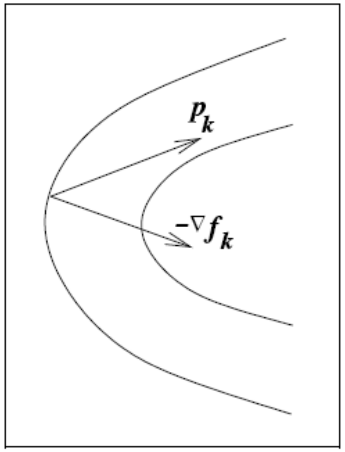
✓any descent direction is guaranteed to produce a decrease in f , provided that the step length is sufficiently small



Descent Methods

Descent methods

✓any descent direction is guaranteed to produce a decrease in f , provided that the step length is sufficiently small



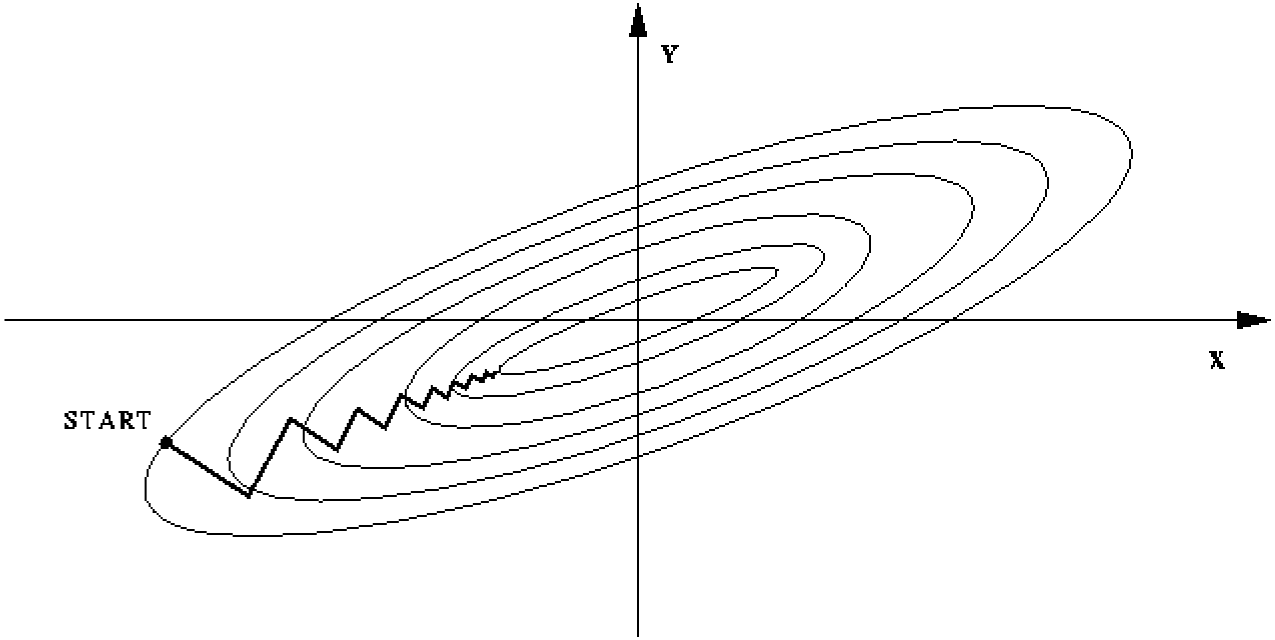
steepest descent direction

✓steepest descent direction $-\nabla f_k$ is the most obvious choice for search direction for a line search method.

✓choose the step length α_k in a variety of ways

steepest descent direction

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



Newton direction

Newton direction

second-order Taylor series approximation

finding the vector p that minimizes $m_k(p)$

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \tfrac{1}{2} p^T \nabla^2 f_k p \overset{\text{def}}{=} m_k(p).$$

Newton direction

Newton direction

second-order Taylor series approximation

finding the vector p that minimizes $m_k(p)$

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \overset{\text{def}}{=} m_k(p).$$

$$p_k^{\mathrm{N}} = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k.$$

The Newton direction is reliable when the difference between the true function $f(x_k + p)$ and its quadratic model $m_k(p)$ is not too large.

Newton direction

The Newton direction is a descent direction

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

✓Fast rate of convergence (quadratic)
✓Main drawback need for the Hessian

Newton direction

The Newton direction is a descent direction

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

Quasi-Newton search direction

✓Fast rate of convergence (quadratic)

✓Main drawback need for the Hessian

Quasi-Newton search directions

✓Quasi-Newton search directions do not require computation of the Hessian

✓still attain a superlinear rate of convergence

✓In place of the true Hessian , *they use an approximation* B_k

Quasi-Newton search directions

✓**Quasi-Newton search directions do not require computation of the Hessian**

✓**still attain a superlinear rate of convergence**

✓**In place of the true Hessian ,** ***they use an approximation B_k***

Hessian approximation

$$s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k.$$

symmetric-rank-one (SR1)

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

Quasi-Newton search directions

✓Quasi-Newton search directions do not require computation of the Hessian

✓still attain a superlinear rate of convergence

✓In place of the true Hessian , *they use an approximation B_k*

Hessian approximation

$$s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k.$$

symmetric-rank-one (SR1)

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

BFGS

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

Quasi-Newton search directions

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

$$H_k \overset{\text{def}}{=} B_k^{-1}$$

(BFGS) $$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \qquad \rho_k = \frac{1}{y_k^T s_k}$$

(SR1) $$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

$$\boxed{p_k = -H_k \nabla f_k}$$

Nonlinear conjugate gradient direction

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1}.$$

✓NLCG more effective than the steepest descent direction

✓almost as simple to compute

✓not attain the fast convergence rates of Newton or quasi-Newton methods

✓not requiring storage of matrices.

Models for Trust Region Methods

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \tfrac{1}{2} p^T B_k p,$$

$B_k = 0$

$$\boxed{\min_p \; f_k + p^T \nabla f_k \qquad \text{subject to } \|p\|_2 \le \Delta_k.}$$

Models for Trust Region Methods

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \tfrac{1}{2} p^T B_k p,$$

$B_k = 0$

$$\min_p \; f_k + p^T \nabla f_k \qquad \text{subject to } \|p\|_2 \le \Delta_k.$$

$$p_k = -\frac{\Delta_k \nabla f_k}{\|\nabla f_k\|}.$$

simply a steepest descent step

Models for Trust Region Methods

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \tfrac{1}{2} p^T B_k p,$$

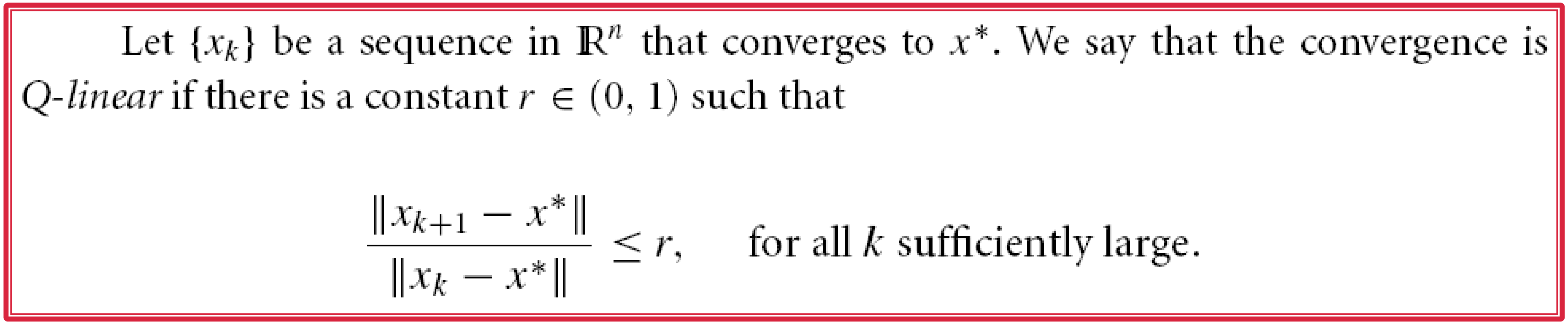
choosing B_k *to be the* exact Hessian $\nabla^2 f_k$

trust-region Newton method

$$\min_p \; m_k(x_k + p), \qquad \text{where } x_k + p \text{ lies inside the trust region.} \qquad \|p\|_2 \leq \Delta_k$$

Rate of Convergence

One of the key measures of performance of an algorithm

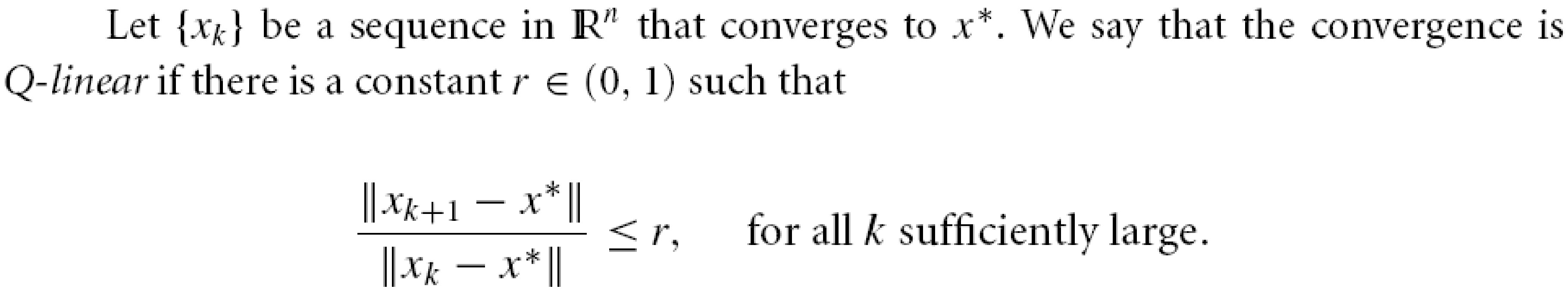


Let $\{x_k\}$ be a sequence in $\mathbb{R}^n$ that converges to x^*. We say that the convergence is *Q-linear* if there is a constant $r \in (0, 1)$ such that

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq r, \qquad \text{for all } k \text{ sufficiently large.}$$

Rate of Convergence

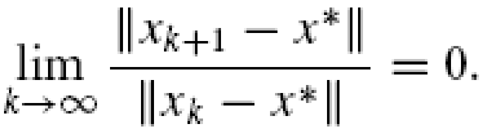
One of the key measures of performance of an algorithm



Let $\{x_k\}$ be a sequence in $\mathbb{R}^n$ that converges to x^*. We say that the convergence is *Q-linear* if there is a constant $r \in (0, 1)$ such that

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \le r, \qquad \text{for all } k \text{ sufficiently large.}$$

The convergence is said to be superlinear if



$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Rate of Convergence

quadratic convergence is obtained if

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M, \qquad \text{for all } k \text{ sufficiently large}$$

Rate of Convergence

quadratic convergence is obtained if

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \le M, \qquad \text{for all } k \text{ sufficiently large}$$

order of convergence is p *(with p > 1) if there is a positive constant M such that*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} \le M, \qquad \text{for all } k \text{ sufficiently large.}$$

Step Length Selection

Line Search Methods

Step length

direction

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

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

choice of the step-length parameter α_k

Step Length Selection

Tradeoff

✓a substantial reduction of *f*

✓Not spend too much time making the choice

too expensive to identify

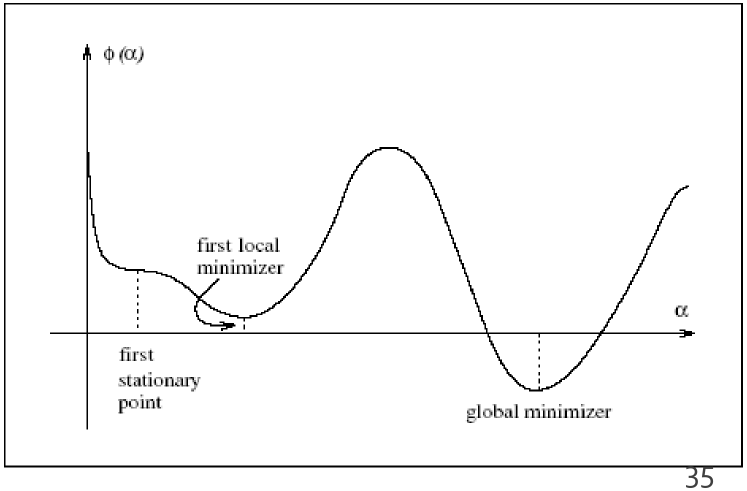
ideal choice would be the global minimizer of

$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0$$

Inexact line search

✓Try out a sequence of candidate values for *α*

✓Stop when certain conditions are satisfied.



Wolfe Conditions

1. sufficient decrease condition

$$\boxed{f(x_k + \alpha p_k) \le f(x_k) + c_1 \alpha \nabla f_k^T p_k}$$

$$\phi(\alpha) \le l(\alpha)$$

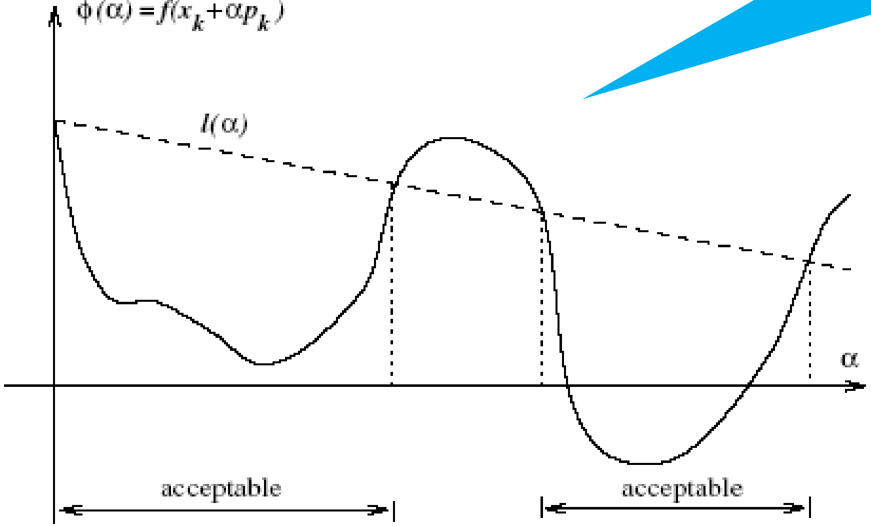
Wolfe Conditions

1. sufficient decrease condition

$$\boxed{f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k}$$

$$\phi(\alpha) \leq l(\alpha)$$

not enough to ensure reasonable progress



Wolfe Conditions

2. curvature condition

$$\boxed{\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k,} \qquad c_2 \in (c_1, 1)$$

$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0$$

Wolfe Conditions

2. curvature condition $\phi'(\alpha_k)$ $\phi'(0)$

$$\boxed{\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k,} \qquad c_2 \in (c_1, 1)$$

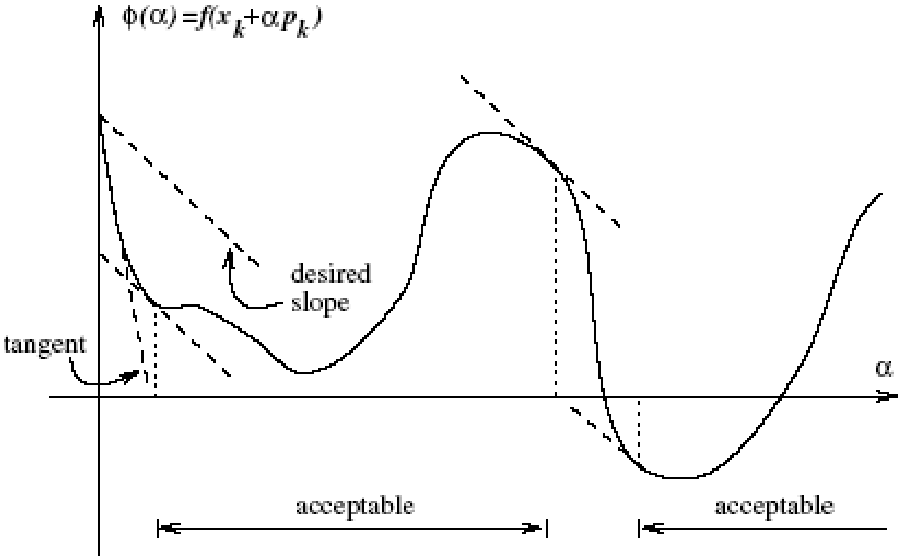
$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0$$

Wolfe Conditions

2. curvature condition $\phi'(\alpha_k)$ $:\phi'(0)$

$$\boxed{\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k,}$$ $: c_2 \in (c_1, 1)$

$$\phi(\alpha) = f(x_k + \alpha p_k), \quad \alpha > 0$$

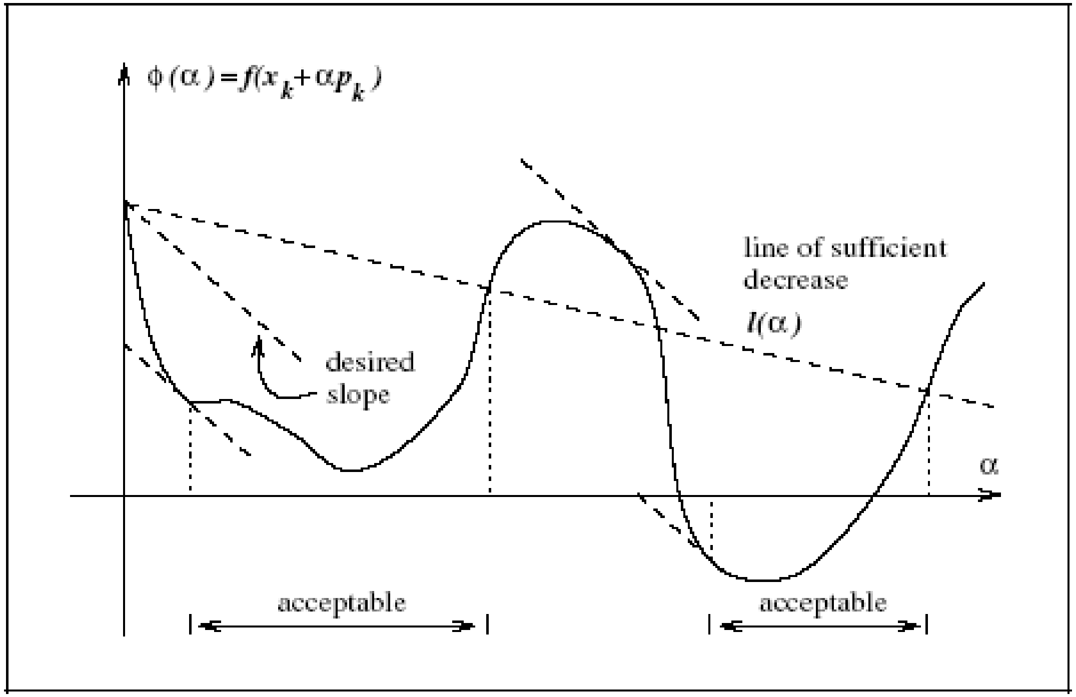


Wolf Conditions

Wolfe Conditions

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

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

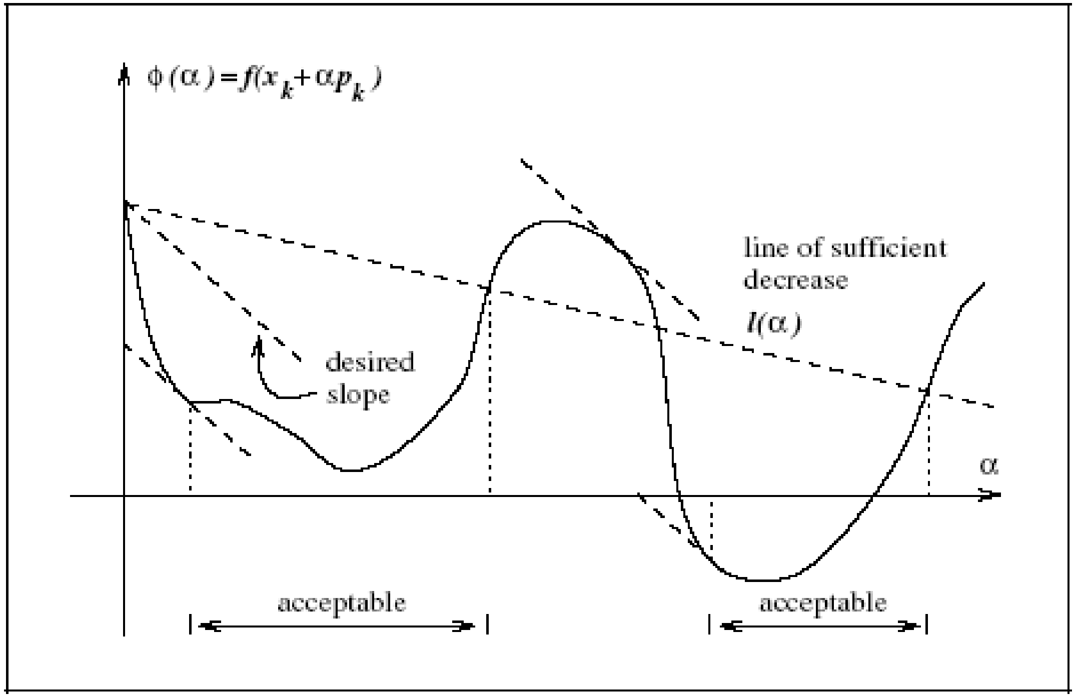


$\phi(\alpha) = f(x_k + \alpha p_k)$
line of sufficient decrease
$l(\alpha)$
desired slope
α
acceptable
acceptable

Wolf Conditions

Wolfe Conditions

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f_k^T p_k,$$
$$\nabla f(x_k + \alpha_k p_k)^T p_k \ge c_2 \nabla f_k^T p_k,$$

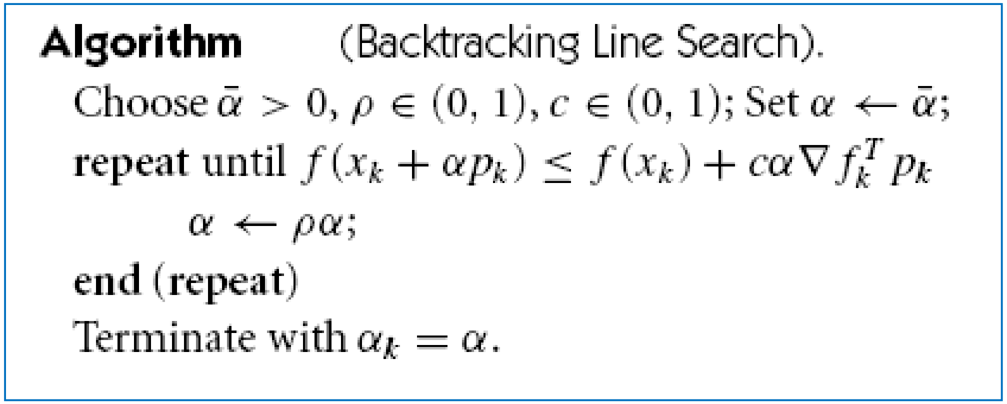


$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f_k^T p_k,$$
$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \le c_2 |\nabla f_k^T p_k|,$$

there exist step lengths that satisfy the Wolfe conditions for every function *f that is smooth and bounded below*

Backtracking

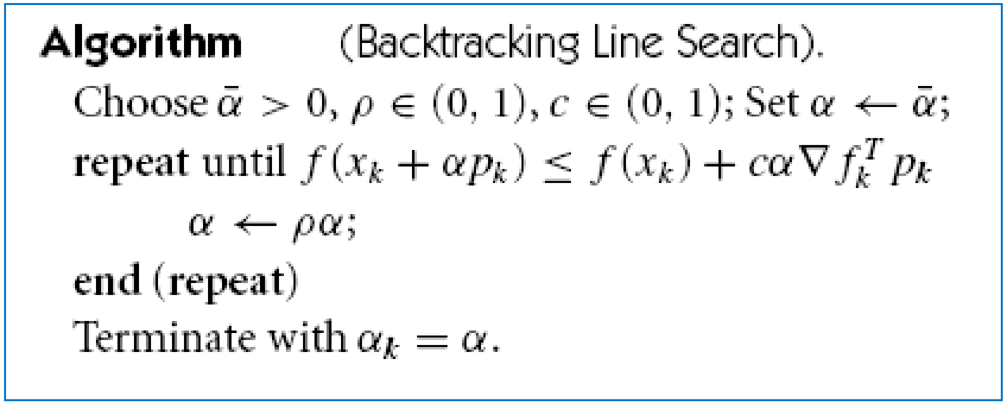
If line search algorithm chooses its candidate step lengths by *backtracking approach*, sufficient decrease condition is sufficient to terminate the line search procedure



```
Algorithm (Backtracking Line Search).
  Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;
  repeat until $f(x_k + \alpha p_k) \le f(x_k) + c\alpha \nabla f_k^T p_k$
      $\alpha \leftarrow \rho\alpha$;
  end (repeat)
  Terminate with $\alpha_k = \alpha$.
```

Backtracking

If line search algorithm chooses its candidate step lengths by *backtracking approach*, sufficient decrease condition is sufficient to terminate the line search procedure



Algorithm (Backtracking Line Search).

```
Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;
repeat until $f(x_k + \alpha p_k) \le f(x_k) + c\alpha \nabla f_k^T p_k$
        $\alpha \leftarrow \rho\alpha$;
end (repeat)
Terminate with $\alpha_k = \alpha$.
```

An acceptable step length will be found after a finite number of trials

selected step length is short enough to satisfy the sufficient decrease condition but not too short.

Interpolation

The aim is to find a value of α that satisfies the sufficient decrease condition without being "too small."

Procedures generate a decreasing sequence of values *αi*

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

$$\boxed{\phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0)}$$

Interpolation

The aim is to find a value of α that satisfies the sufficient decrease condition without being “too small.”

Procedures generate a decreasing sequence of values *αi*

$$f(x_k + \alpha p_k) \le f(x_k) + c_1 \alpha \nabla f_k^T p_k$$

$$\boxed{\phi(\alpha_k) \le \phi(0) + c_1 \alpha_k \phi'(0)}$$

α_0 is given. If we have

$$\phi(\alpha_0) \le \phi(0) + c_1 \alpha_0 \phi'(0)$$

terminate the search

Interpolation

interval $[0, \alpha0]$ contains acceptable step lengths

✓form a quadratic approximation ϕ by interpolating the three pieces of information available—$\phi(0)$, $\phi'(0)$, and $\phi(\alpha0)$

Interpolation

interval [o, αo] contains acceptable step lengths

✓form a quadratic approximation ϕ by interpolating the three pieces of information available—ϕ(o), ϕ’(o), and ϕ(αo)

$$\phi_q(\alpha) = \left(\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2} \right) \alpha^2 + \phi'(0)\alpha + \phi(0)$$

✓The new trial value α1 is defined as the minimizer of this quadratic

Interpolation

interval [o, αo] contains acceptable step lengths

✓form a quadratic approximation ϕ by interpolating the three pieces of information available—ϕ(o), ϕ'(o), and ϕ(αo)

$$\phi_q(\alpha) = \left(\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2} \right) \alpha^2 + \phi'(0)\alpha + \phi(0)$$

✓The new trial value α1 is defined as the minimizer of this quadratic

$$\alpha_1 = -\frac{\phi'(0)\alpha_0^2}{2\left[\phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0\right]}$$

✓If the sufficient decrease condition satisfied at *α1, we terminate the search.*

Interpolation

✓we construct a cubic function that interpolates the four pieces of information ϕ(0), ϕ(0), ϕ(α0), and ϕ(α1)

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \alpha\phi'(0) + \phi(0)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_0^2\alpha_1^2(\alpha_1 - \alpha_0)} \begin{bmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_1) - \phi(0) - \phi'(0)\alpha_1 \\ \phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0 \end{bmatrix}$$

Interpolation

✓we construct a cubic function that interpolates the four pieces of information ϕ(0), ϕ(0), ϕ(α0), and ϕ(α1)

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \alpha\phi'(0) + \phi(0)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_0^2\alpha_1^2(\alpha_1 - \alpha_0)} \begin{bmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_1) - \phi(0) - \phi'(0)\alpha_1 \\ \phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0 \end{bmatrix}$$

✓the minimizer α2 of ϕc

$$\alpha_2 = \frac{-b + \sqrt{b^2 - 3a\phi'(0)}}{3a}$$

✓If the sufficient decrease condition satisfied at *α2, we terminate the search.*

✓this process is repeated, using a cubic interpolant of ϕ(0), ϕ'(0) and the two most recent values of ϕ, until an α that satisfies is located

Initial Step Length

$\alpha_0 = 1$ → For Newton and quasi-Newton methods

first-order change in the function at iterate *xk will be the same as that obtained at the previous* step

$$\alpha_0 = \alpha_{k-1} \frac{\nabla f_{k-1}^T p_{k-1}}{\nabla f_k^T p_k}.$$

Trust Region Methods

Trust-region methods

- Trust-region methods define a region within *trust* the model
- choose the step to be the minimizer of the model in this region
- choose the direction and length of the step simultaneously
- If a step is not acceptable, reduce the size of the region
- size of the trust region is critical
- performance of the algorithm during previous iterations

Trust-region methods

$$m_k(p) = f_k + g_k^T p + \tfrac{1}{2} p^T B_k p$$

$$f_k = f(x_k) \text{ and } g_k = \nabla f(x_k)$$

$$\boxed{\min_{p \in \mathrm{R}^n} m_k(p) = f_k + g_k^T p + \tfrac{1}{2} p^T B_k p \qquad \text{s.t. } \|p\| \le \Delta_k}$$

Trust-region radius

Base this choice on the agreement between the model function *mk* and the objective function *f* at previous iterations

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$$

Trust-region radius

Base this choice on the agreement between the model function *mk* and the objective function *f* at previous iterations

$$\rho_k = \frac{\overbrace{f(x_k) - f(x_k + p_k)}^{\text{Actual Reduction}}}{\underbrace{m_k(0) - m_k(p_k)}_{\text{Predicted Reduction}}}$$

- Predicted reduction nonnegative

Trust-region radius

Base this choice on the agreement between the model function *mk* and the objective function *f* at previous iterations

Actual Reduction

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$$

Predicted Reduction

- Predicted reduction nonnegative
- Ratio
  - is negative ⇨ step rejected
  - close to 1 ⇨ good agreement
  - positive but significantly smaller than 1 ⇨ not change in radious
  - Close to zero or negative ⇨ reduce radious

Trust-region radius

```
Algorithm (Trust Region).
Given $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in \left[0, \frac{1}{4}\right)$:
for $k = 0, 1, 2, \ldots$
        Obtain $p_k$ by (approximately) solving $\min_{p \in \mathbb{R}^n} m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$   s.t. $\|p\| \le \Delta_k$
        Evaluate $\rho_k$
        if $\rho_k < \frac{1}{4}$
                $\Delta_{k+1} = \frac{1}{4}\Delta_k$
        else
                if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$
                        $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$
                else
                        $\Delta_{k+1} = \Delta_k$;
        if $\rho_k > \eta$
                $x_{k+1} = x_k + p_k$
        else
                $x_{k+1} = x_k$;
end (for).
```

Trust-region methods

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$$

$f_k = f(x_k)$ and $g_k = \nabla f(x_k)$

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \qquad \text{s.t. } \|p\| \le \Delta_k$$

solve a sequence of subproblems

p_k^*

Trust-region methods

$$m_k(p) = f_k + g_k^T p + \tfrac{1}{2} p^T B_k p$$

$f_k = f(x_k)$ and $g_k = \nabla f(x_k)$

$$\boxed{\min_{p \in \mathrm{R}^n} m_k(p) = f_k + g_k^T p + \tfrac{1}{2} p^T B_k p \qquad \text{s.t. } \|p\| \le \Delta_k}$$

p_k^*

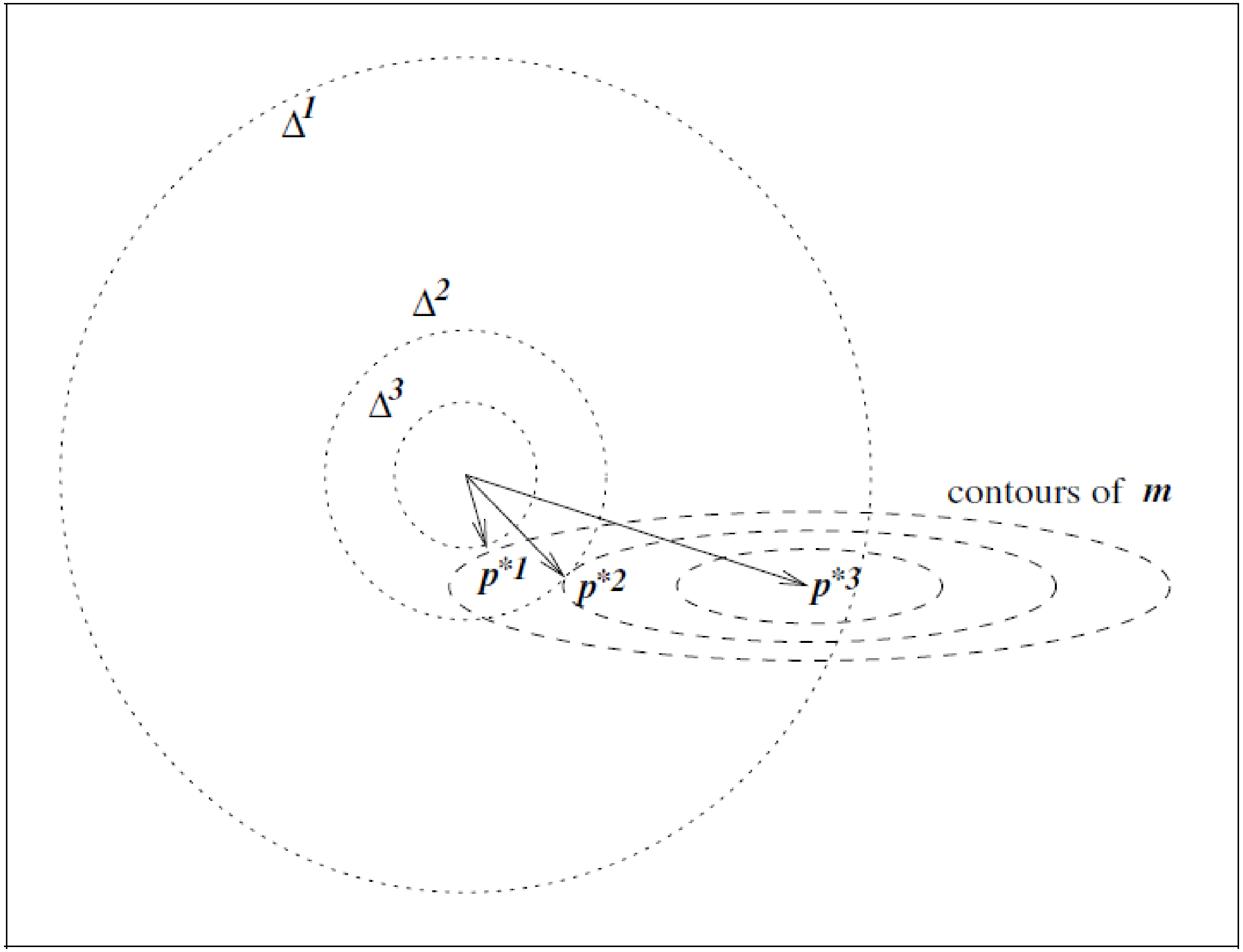
solve a sequence of subproblems

When B_k is positive definite and $\|B_k^{-1} g_k\| \le \Delta_k$

$$p_k^{\mathrm{B}} = -B_k^{-1} g_k$$

✓ The solution of is not so obvious in other cases need only an *approximate* solution to obtain convergence and good practical behavior

Trust-region



Cauchy Point

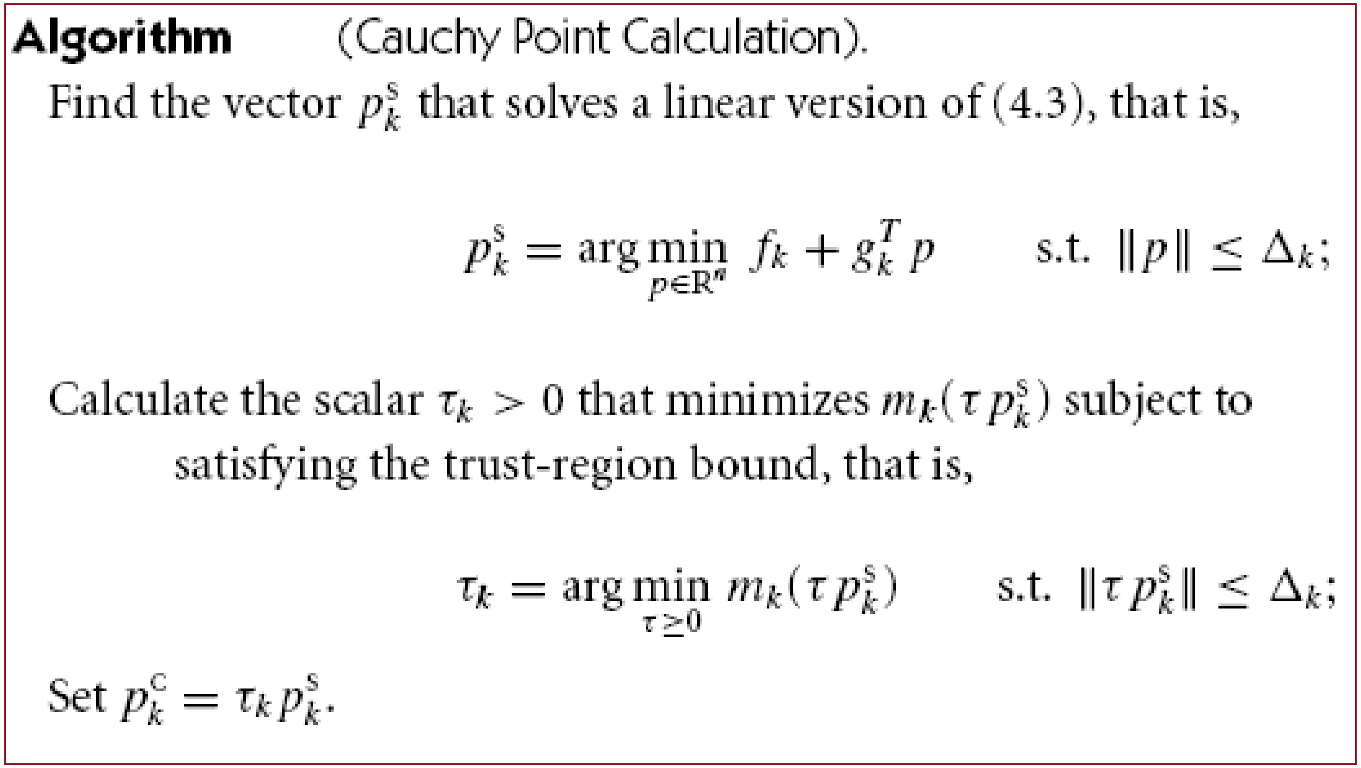
- line search methods can be globally convergent even when the optimal step length is not used at each iteration
- Seek the optimal solution of the subproblem
- for global convergence to find an approximate solution p_k lies in the trust region and gives a *sufficient reduction* in the model
- sufficient reduction can be quantified in terms of the Cauchy point p_k^{c}

Cauchy Point

- line search methods can be globally convergent even when the optimal step length is not used at each iteration
- Seek the optimal solution of the subproblem
- for global convergence to find an approximate solution p_k lies in the trust region and gives a *sufficient reduction* in the model
- sufficient reduction can be quantified in terms of the Cauchy point p_k^{C}

a trust-region method will be globally convergent if its steps p_k *give a reduction in the model* m_k *that is at least some fixed positive* multiple of the decrease attained by the Cauchy step.

Cauchy Point



Algorithm (Cauchy Point Calculation).

Find the vector p_k^s that solves a linear version of (4.3), that is,

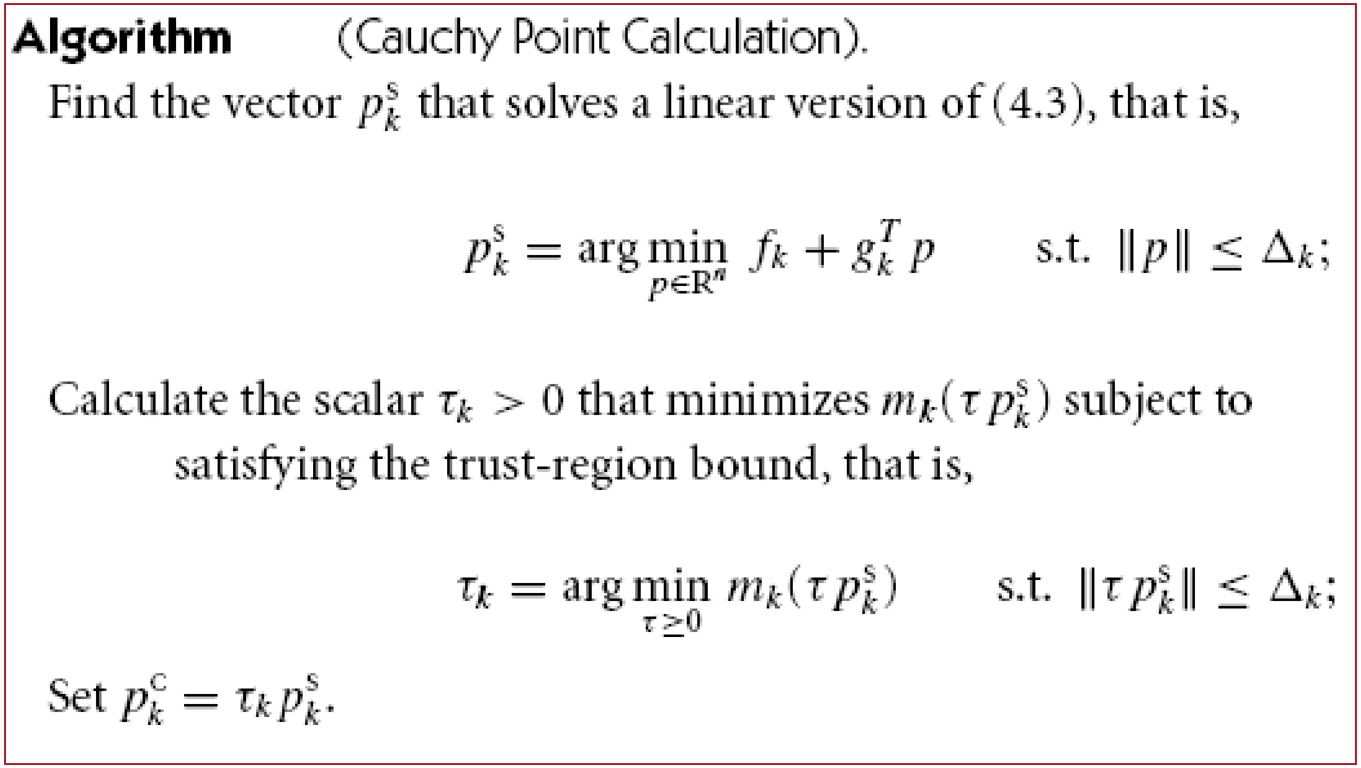
$$p_k^s = \arg\min_{p \in \mathbb{R}^n} f_k + g_k^T p \qquad \text{s.t. } \|p\| \le \Delta_k;$$

Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^s)$ subject to satisfying the trust-region bound, that is,

$$\tau_k = \arg\min_{\tau \ge 0} m_k(\tau p_k^s) \qquad \text{s.t. } \|\tau p_k^s\| \le \Delta_k;$$

Set $p_k^C = \tau_k p_k^s$.

Cauchy Point



Algorithm (Cauchy Point Calculation).

Find the vector p_k^{s} that solves a linear version of (4.3), that is,

$$p_k^{\mathrm{s}} = \arg\min_{p \in \mathbb{R}^n} f_k + g_k^T p \qquad \text{s.t. } \|p\| \le \Delta_k;$$

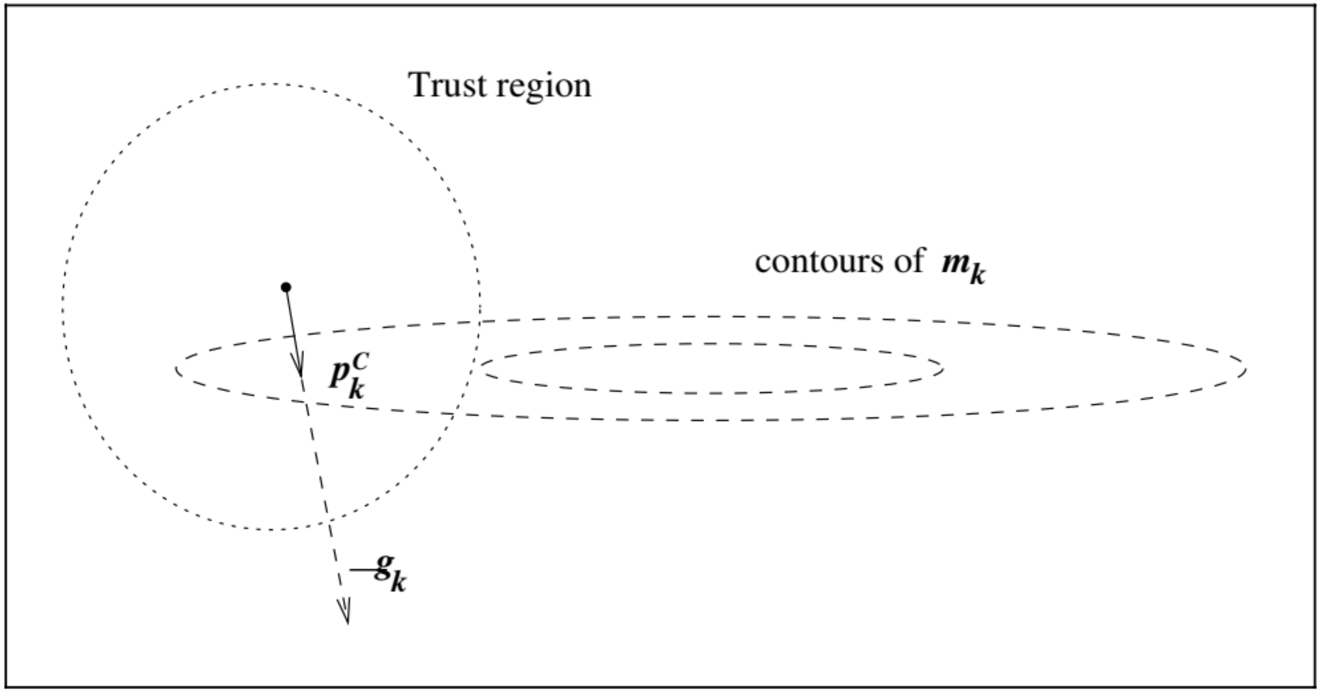
Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^{\mathrm{s}})$ subject to satisfying the trust-region bound, that is,

$$\tau_k = \arg\min_{\tau \ge 0} m_k(\tau p_k^{\mathrm{s}}) \qquad \text{s.t. } \|\tau p_k^{\mathrm{s}}\| \le \Delta_k;$$

Set $p_k^{\mathrm{C}} = \tau_k p_k^{\mathrm{s}}$.

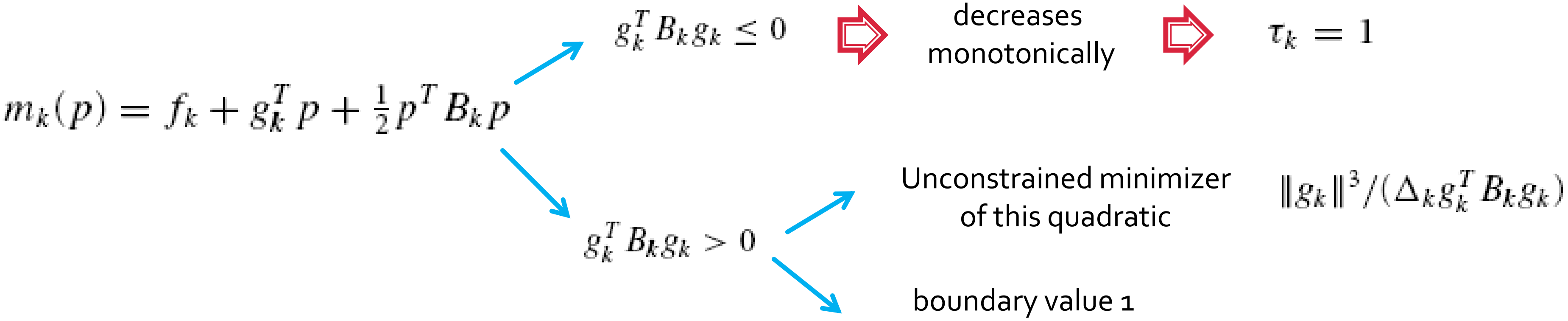
⇨ $$p_k^{\mathrm{s}} = -\frac{\Delta_k}{\|g_k\|} g_k.$$

minimizer of m_k *along the steepest* descent direction $-g_k$ *. subject to the trust-region bound.*


Trust region
contours of m_k
p_k^C
$-g_k$

Cauchy Point

$$p_k^{\text{s}} = -\frac{\Delta_k}{\|g_k\|} g_k. \Rightarrow \tau_k = \arg\min_{\tau \geq 0} m_k(\tau p_k^{\text{s}}) \quad \text{s.t. } \|\tau p_k^{\text{s}}\| \leq \Delta_k; \Rightarrow p_k^{\text{c}} = \tau_k p_k^{\text{s}}$$



$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$

$g_k^T B_k g_k \leq 0$ $\Rightarrow$ decreases monotonically $\Rightarrow$ $\tau_k = 1$

$g_k^T B_k g_k > 0$

Unconstrained minimizer of this quadratic $\|g_k\|^3/(\Delta_k g_k^T B_k g_k)$

boundary value 1

$$p_k^{\text{c}} = -\tau_k \frac{\Delta_k}{\|g_k\|} g_k \qquad \tau_k = \begin{cases} 1 & \text{if } g_k^T B_k g_k \leq 0; \\ \min\left(\|g_k\|^3/(\Delta_k g_k^T B_k g_k), 1\right) & \text{otherwise.} \end{cases}$$

Cauchy Point

Problems?

always taking the Cauchy point as our step?

Cauchy Point

Problems?

always taking the Cauchy point as our step?

Some algorithms for approximate solutions of subproblem

$$\min_{p\in\mathbb{R}^n} m(p) = f + g^T p + \tfrac{1}{2} p^T B p, \qquad \text{s.t. } \|p\| \le \Delta$$

dropping the subscript "*k*" $p^*(\Delta)$

Dogleg Method

B is positive definite.

$$\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \frac{1}{2} p^T B p, \quad \text{s.t. } \|p\| \le \Delta$$

$$p^B = -B^{-1} g$$

Dogleg Method

B is positive definite.

$$\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \tfrac{1}{2} p^T B p, \quad \text{s.t. } \|p\| \le \Delta$$

$$p^B = -B^{-1} g$$

$$\boxed{p^*(\Delta) = p^B, \qquad \text{when } \Delta \ge \|p^B\|}$$

Dogleg Method

B is positive definite.

$$\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \tfrac{1}{2} p^T B p, \qquad \text{s.t. } \|p\| \le \Delta$$

$$p^B = -B^{-1} g$$

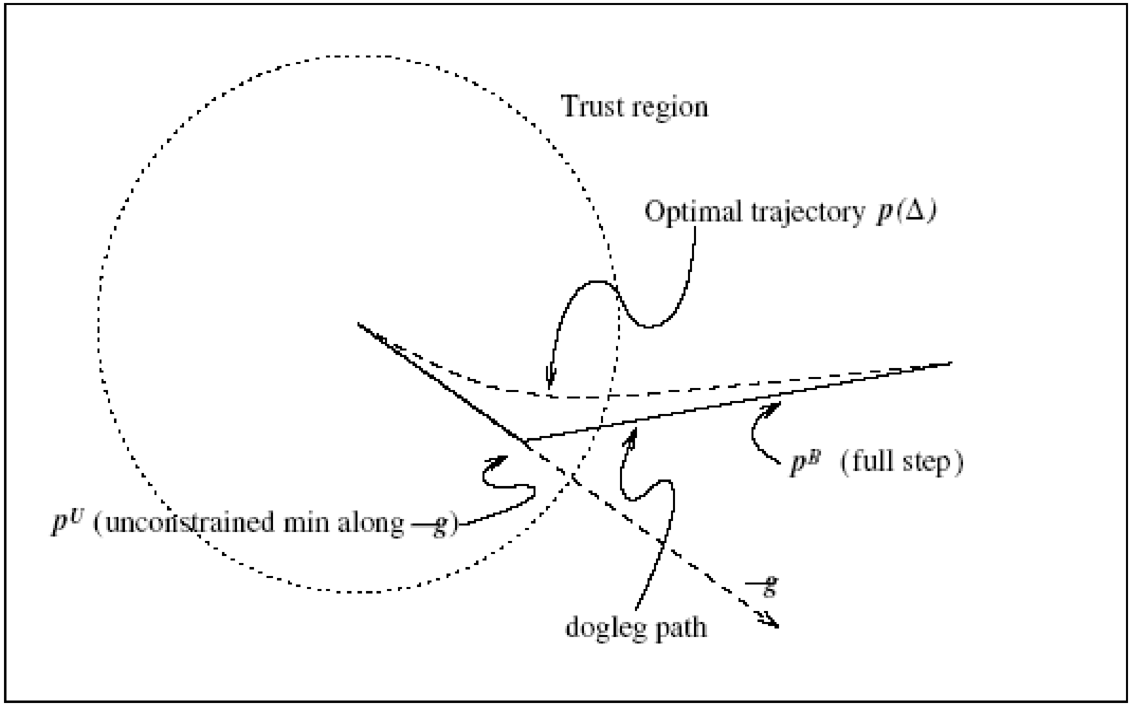
$$\boxed{p^*(\Delta) = p^B, \qquad \text{when } \Delta \ge \|p^B\|}$$

Δ is small omitting the quadratic term

$$\boxed{p^*(\Delta) \approx -\Delta \frac{g}{\|g\|}, \qquad \text{when } \Delta \text{ is small.}}$$

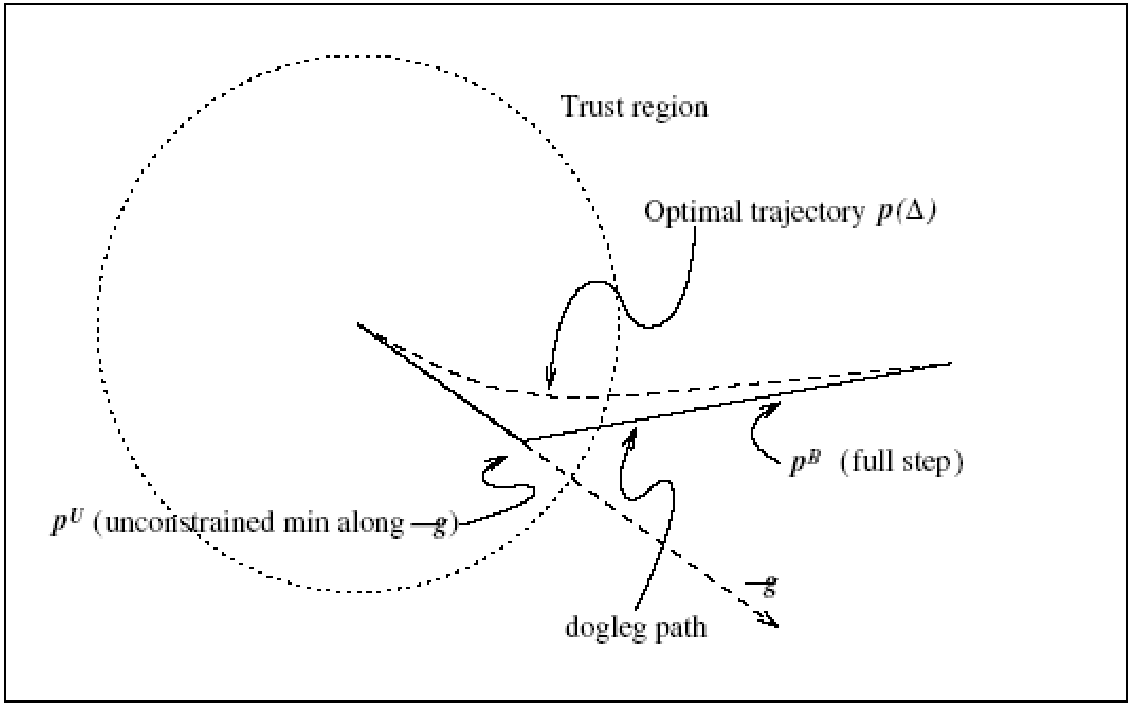
For intermediate values of , the solution *p ∗(Δ) typically follows a curved trajectory*

Dogleg Method


Trust region
Optimal trajectory $p(\Delta)$
p^B (full step)
p^U (unconstrained min along $-g$)
$-g$
dogleg path

Idea of dogleg method: replacing the curved trajectory for *p∗(Δ) with a path consisting of two line segments.*

Dogleg Method


Trust region
Optimal trajectory $p(\Delta)$
p^B (full step)
p^U (unconstrained min along $-g$)
$-g$
dogleg path

Idea of dogleg method: replacing the curved trajectory for *p∗(Δ) with a path consisting of two line segments.*

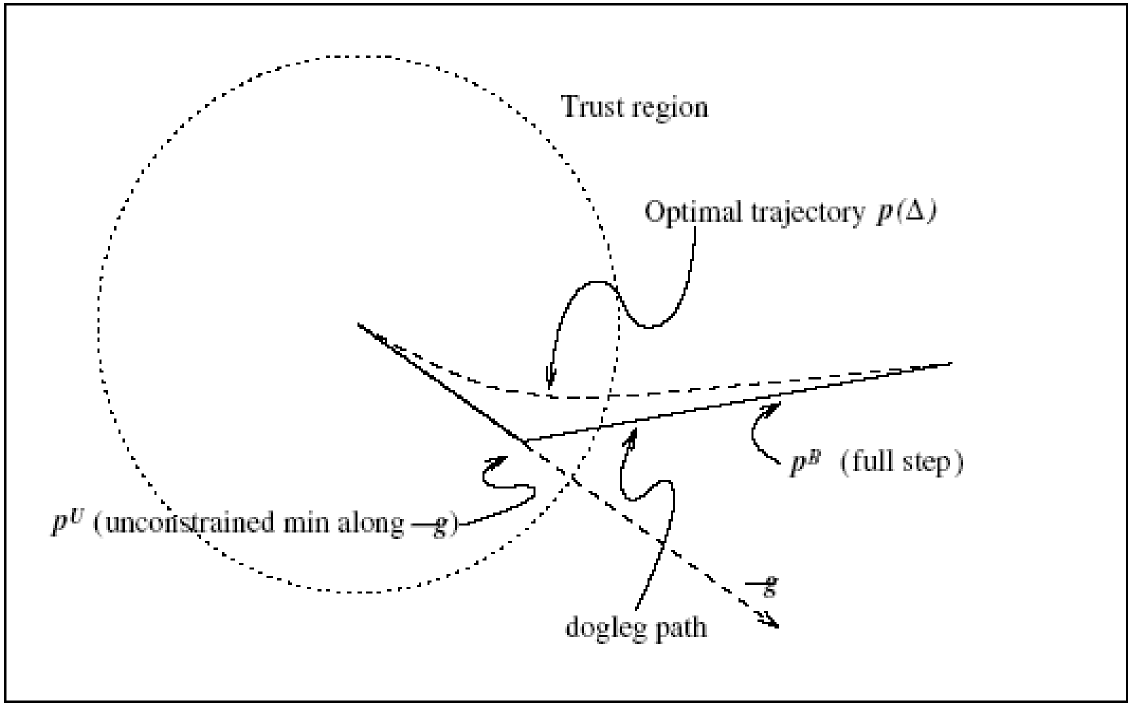
✓first line segment runs from the origin to the minimizer of *m along the steepest descent direction*

✓second line segment runs from p_U *to* p_B

Dogleg Method

$$\tilde{p}(\tau) = \begin{cases} \tau p^{U}, & 0 \le \tau \le 1, \\ p^{U} + (\tau - 1)(p^{B} - p^{U}), & 1 \le \tau \le 2. \end{cases}$$

$$p^{U} = -\frac{g^T g}{g^T B g} g$$


Trust region
Optimal trajectory $p(\Delta)$
p^B (full step)
p^U (unconstrained min along $-g$)
$-g$
dogleg path

Idea of dogleg method: replacing the curved trajectory for *$p\ast(\Delta)$ with a path consisting of two line segments.*

✓first line segment runs from the origin to the minimizer of *m along the steepest descent direction*

✓second line segment runs from p_U *to* p_B

Dogleg Method

- The dogleg method chooses *p to minimize the model m along this path, subject to* the trust-region bound.
- This line intersects the trust-region boundary *at exactly one point if* $\|p^{\mathrm{B}}\| \geq \Delta$

$$\|p^{\mathrm{U}} + (\tau - 1)(p^{\mathrm{B}} - p^{\mathrm{U}})\|^2 = \Delta^2$$

Two-dimensional subspace minimization

widening the search for *p to the entire two-dimensional subspace spanned* by p_U *and* p_B

$$\boxed{\min_{p} m(p) = f + g^T p + \tfrac{1}{2} p^T B p \quad \text{s.t. } \|p\| \le \Delta, \; p \in \operatorname{span}[g, B^{-1} g].}$$

Two-dimensional subspace minimization

widening the search for p *to the entire two-dimensional subspace spanned* by p_U *and* p_B

$$\min_p m(p) = f + g^T p + \tfrac{1}{2} p^T B p \quad \text{s.t. } \|p\| \le \Delta, \; p \in \operatorname{span}[g, B^{-1} g].$$

Cauchy point p_C *is feasible*

✓optimal solution of this subproblem yields at least as much reduction in m *as the Cauchy point*

✓*global convergence* of the algorithm

✓extension of the dogleg method entire dogleg path lies in span[g, $B^{-1}g$].

Two-dimensional subspace minimization

widening the search for *p to the entire two-dimensional subspace spanned* by p_U *and* p_B

$$\boxed{\min_p m(p) = f + g^T p + \tfrac{1}{2} p^T B p \quad \text{s.t. } \|p\| \le \Delta, \; p \in \operatorname{span}[g, B^{-1}g].}$$

Cauchy point p_C *is feasible*

✓optimal solution of this subproblem yields at least as much reduction in *m as the Cauchy point*

✓*global convergence* of the algorithm

✓extension of the dogleg method entire dogleg path lies in span[*g, B*$^{-1}$*g].*

Two-dimensional subspace minimization

When *B has negative eigenvalues* $\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \frac{1}{2} p^T B p, \quad \text{s.t. } \|p\| \leq \Delta$

$$\min_p m(p) = f + g^T p + \tfrac{1}{2} p^T B p \quad \text{s.t. } \|p\| \leq \Delta, \qquad \operatorname{span}[g, (B + \alpha I)^{-1} g],$$