Nonlinear Optimization

Chapter 4: Unconstrained Nonlinear Optimization

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on Beep "[2019-20] - Optimization"

Academic year 2019-20

4.1 Examples

1) Statistical estimation

Random variable X with density $f(x, \underline{\theta})$, where $\underline{\theta} \in \mathbb{R}^m$ is parameter vector, and independent observations $x_1, \ldots, x_n$.

Maximum likelihood: Estimates $\hat{\underline{\theta}}$ of $\underline{\theta}$ are derived by maximizing

$$L(\underline{\theta}) = f(x_1, \underline{\theta})\, f(x_2, \underline{\theta}) \ldots f(x_n, \underline{\theta})$$

Assumption: $\exists\, \underline{\theta}$ for which all factors are positive.

Since ln(.) is monotonically increasing, $\hat{\underline{\theta}}$ also maximizes

$$\ln(L(\underline{\theta})) = \sum_{j=1}^{n} \ln(f(x_j, \underline{\theta}))$$

If f is differentiable w.r.t. $\underline{\theta}$ at $\hat{\underline{\theta}}$, necessary optimality conditions:

$$\sum_{j=1}^{n} \frac{\nabla_{\underline{\theta}} f(x_j, \hat{\underline{\theta}})}{f(x_j, \hat{\underline{\theta}})} = \underline{0}$$

(stationarity conditions)

For Guassian density

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2\sigma^2}$$

and $\underline{\theta} = (\mu, \sigma)$, we obtain

$$\ln(L(\underline{\theta})) = \ln((\frac{1}{\sigma\sqrt{2\pi}})^n \prod_{j=1}^{n} \exp - \frac{(x_j-\mu)^2}{2\sigma^2}) = -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2$$

Minimum is achieved in a stationary point:

$$\frac{\partial[\ln(L(\underline{\theta}))]}{\partial\mu} = \frac{1}{\sigma^2}\sum_{j=1}^{n}(x_j-\mu) = 0$$

and

$$\frac{\partial[\ln(L(\underline{\theta}))]}{\partial\sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{n}(x_j-\mu)^2 = 0$$

Thus

$$\hat{\mu} = \frac{1}{n}\sum_{j=1}^{n} x_j \qquad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_j-\hat{\mu})^2}$$

(MLE)

2) Training multilayer neural networks

Supervised learning:

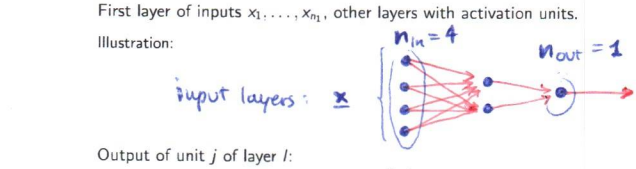
Given a training set $T = \{(\underline{x}^1, \underline{y}^1), \ldots, (\underline{x}^p, \underline{y}^p)\}$ where $\underline{y}^k \in [0,1]^{n_{out}}$ desiderd output for $\underline{x}^k \in \mathbb{R}^{n_{in}}$, construct a model that maps $\underline{x}^k$'s into $\underline{y}^k$'s as well as possible.

Multilayer networks:

L layers with n_l units in layer l, $n_1 = n_{in}$ and $n_L = n_{out}$.

First layer of inputs $x_1, \ldots, x_{n_1}$, other layers with activation units.

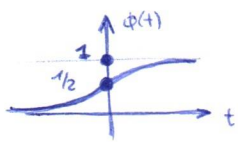
Illustration:



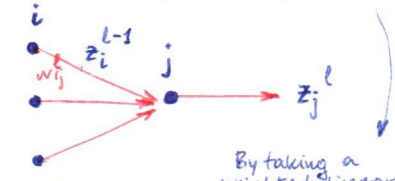
Output of unit j of layer l:

$$z_j^l = \phi(\sum_{i=1}^{n_{l-1}} w_{ij}^l z_i^{l-1} - w_{0j}^l)$$

where weights w_{ij} to be determined and $\phi : \mathbb{R} \to \mathbb{R}$ is sigmoid $\phi(t) = \frac{1}{1+e^{-t}}$.

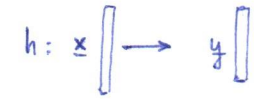


we call "activation unit" all the dots except the inputs.
consider the j-th unit in the layer l.
How this unit compute the output?



By taking a weighted linear comb. of the inputs minus a threshold

A multilayer network defnines a mapping $h(\underline{w}, .)$ from $\mathbb{R}^{n_1}$ to $\mathbb{R}^{n_L}$ parametrized by $\underline{w} = \{w_{ij}^l : l = 1, \dots, L; i = 1, \dots, n_{l-1}; j = 1, \dots, n_l\}$.



Training problem: Given $T = \{(\underline{x}^1, \underline{y}^1), \dots, (\underline{x}^P, \underline{y}^P)\}$, select appropriate values of $\underline{w}$ to approximate as well as possible the mapping underlying T.

In general one minimizes (least squares)

$$\frac{1}{2}\sum_{k=1}^{P}(\|\underline{y}^k - h(\underline{w}, \underline{x}^k)\|)^2$$

this is a unconstrained nonlinear optimization problem

↳ we assume that the training set is extracted from a given mapping that we would like to learn

Quite challenging, typically nonconvex with multiple local minima.

Illustration:

4.2 Optimality conditions

Generic optimization problem:

$$\min_{\underline{x} \in S} f(\underline{x})$$

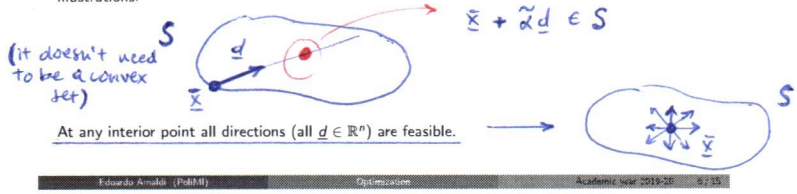
where $S \subseteq \mathbb{R}^n$, $f : S \to \mathbb{R}$ and $f \in C^1$ or C^2. (once or twice continuously differentiable)

(Unconstrained case: $S = \mathbb{R}^n$)

Definition: $\underline{d} \in \mathbb{R}^n$ is a *feasible direction* at $\overline{\underline{x}}$ if

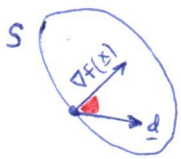
$$\exists\, \overline{\alpha} > 0 \text{ such that } \overline{\underline{x}} + \alpha \underline{d} \in S \qquad \forall \alpha \in [0, \overline{\alpha}] \qquad (1)$$

Illustrations:



(it doesn't need to be a convex set)

At any interior point all directions (all $\underline{d} \in \mathbb{R}^n$) are feasible.



◀ ≤ 180°

⟹ all the feasible directions $\underline{d}$ are ascent directions

First order necessary local optimality conditions:

If $f \in C^1$ on S and $\overline{\underline{x}}$ is a *local minimum* of f over S, then for any feasible direction $\underline{d} \in \mathbb{R}^n$ at $\overline{\underline{x}}$

$$\nabla^t f(\overline{\underline{x}})\underline{d} \geq 0,$$

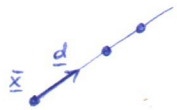
namely all feasible directions are ascent directions.

Proof:

According to (1), consider

$$\phi : [0, \overline{\alpha}] \to \mathbb{R} \text{ such that } \phi(\alpha) = f(\overline{\underline{x}} + \alpha \underline{d})$$

Since $\overline{\underline{x}}$ is a local minimum of f over S, $\alpha = 0$ is a local minimum of $\phi(\alpha)$.



Taylor series of ϕ at $\alpha = 0$

$$\phi(\alpha) = \phi(0) + \alpha\phi'(0) + o(\alpha)$$

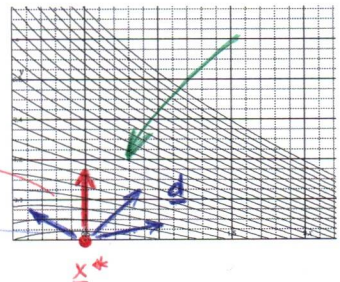
N.B.: $u(\alpha) = o(\alpha)$ if $u(\alpha)$ tends to 0 faster than α when $\alpha \to 0$.

Suppose $\phi'(0) < 0$: if $\alpha \to 0$ we neglet $o(\alpha)$ and : $\phi(\alpha) - \phi(0) = \alpha\phi'(0) < 0$, then 0 would not be a local minimum. Therefore $\phi'(0) \geq 0$ and since

$$\phi'(\alpha) = \nabla^T f(\overline{\underline{x}} + \alpha \underline{d})\,\underline{d} \implies \nabla^T f(\overline{\underline{x}} + \alpha \underline{d}) \geq 0 \quad \blacksquare$$

since $\exists \alpha$ s.t. the value of ϕ in α is strictly smaller

Example: $\min_{x_1,\, x_2 \geq 0} f(x_1, x_2) = x_1^2 - x_1 + x_2 + x_1 x_2$



these are the level curves, the function is getting smaller in the direction →

$\underline{x}^* = (\frac{1}{2}\ 0)^t$ is a global minimum because $\nabla^t f(\underline{x}^*)\underline{d} \geq \underline{0}$ for all feasible directions $\underline{d}$ at $\underline{x}^*$, even if $\nabla^t f(\underline{x}^*) = (0\ \frac{3}{2}) \neq \underline{0}$.

Second order necessary local optimality conditions:

If $f \in \mathcal{C}^2$ on S and $\bar{x}$ is a *local minimum* of f over S then

i) $\nabla^t f(\bar{x})d \geq 0 \qquad \forall d \in \mathbb{R}^n$ feasible direction at $\bar{x}$,

ii) if $\nabla^t f(\bar{x})d = 0$ then $d^t \nabla^2 f(\bar{x}) d \geq 0$.

← positive def. of the Hessian for all $d \perp \nabla^T f(\bar{x})$

Proof:

We proceed similarly for (ii).

Suppose $\nabla^t f(\bar{x})d = 0$, then

$\phi'(0)$ $\qquad \phi(\alpha) = \phi(0) + \alpha \underbrace{\phi'(0)}_{0} + \frac{1}{2}\alpha^2\phi''(0) + o(\alpha^2).$

If $\phi''(0) = d^t \nabla^2 f(\bar{x}) d < 0$, for sufficiently small values of α we would have

$$\phi(\alpha) - \phi(0) \leq \frac{1}{2}\alpha^2\phi''(0) < 0,$$

namely 0 would not be a local minimum of $\phi(\alpha)$.

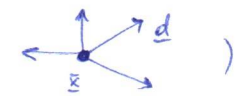
Hence $\phi''(0) = d^t \nabla^2 f(\bar{x}) d \geq 0$. □

Corollary: (Unconstrained case)

If $f \in \mathcal{C}^2$ on S and $\bar{x} \in int(S)$ is a *local minimum* of f over S, then

1. $\nabla f(\bar{x}) = 0$ (stationarity condition)
2. $\nabla^2 f(\bar{x})$ is positive semidefinite.

if S is unconstrained then all the points are interior points (and so any direction is a feasible direction :



)

Proof:

Since $\bar{x} \in int(S)$, all $d \in \mathbb{R}^n$ are feasible directions at $\bar{x}$.
Then $\nabla^t f(\bar{x})d \geq 0$ for every d and $-d$ imply (1).
(2) is an immediate consequence of $d^t \nabla^2 f(\bar{x}) d \geq 0$ for all $d \in \mathbb{R}^n$. □

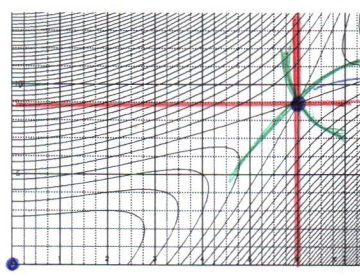
Types of candidate points: local minima, local maxima and saddle points.

points that satisfy these conditions

min in one direction, max in the other one

These optimality conditions are not sufficient.
E.g. $f(x) = x^3$ with $f'(0) = 0$ and $f''(0) = 0$ but $x = 0$ not a local minimum.

Example: $\min_{x_1,\, x_2 \geq 0} f(x_1, x_2) = x_1^3 - x_1^2 x_2 + 2x_2^2$



(6,9) not even a local minimum (saddle point)

How can we tell (that is a candidate)? in the direction $x_1 = 6$ we have that the point is a minimum (convex function). Also in the other direction $x_2 = 9$ the point is a minimum

But the Hessian matrix at the point (6,9) is not pos. semidefinite.

(0,0) Optimal solution

Candidate points: (0 0) and (6 9).
(6 9) not a local minimum even though, for $x_1 = 6$, $x_2 = 9$ it is a local minimum w.r.t. x_2 and, for $x_2 = 9$, $x_1 = 6$ it is a local minimum w.r.t. x_1.

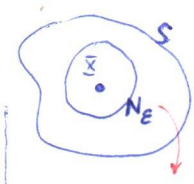
Sufficient local optimality conditions:

Since $\bar{x} \in int(S)$ we don't have to worry about d since $\forall d$ is a feasible direction

If $f \in \mathcal{C}^2$ on S and $\bar{x} \in int(S)$ such that $\nabla f(\bar{x}) = 0$ and $\nabla^2 f(\bar{x})$ is positive definite, then $\bar{x}$ is a *strict local minimum* of f over S, namely

$$f(x) > f(\bar{x}) \qquad \forall x \in \mathcal{N}_\epsilon(\bar{x}) \cap S.$$

→ $d^T \nabla^2 f(\bar{x})\, d > 0 \quad \forall d \neq 0$



$N_\epsilon(\bar{x}) = \{ x \in \mathbb{R}^n : \| x - \bar{x} \| < \epsilon \}$

Proof:

Let $d \in \mathcal{B}_\epsilon(0)$ be any feasible direction such that $\bar{x} + d \in S \cap \mathcal{B}_\epsilon(\bar{x})$.

Then

$$f(\bar{x} + d) = f(\bar{x}) + \underbrace{\nabla^t f(\bar{x})\, d}_{0} + \frac{1}{2} d^t \nabla^2 f(\bar{x}) d + o(\|d\|^2)$$

Since $\nabla^2 f(\bar{x})$ is positive def. $\Rightarrow \exists\, a > 0 : d^T \nabla^2 f(\bar{x})\, d \geq a \|d\|^2$ where a is the smallest eigenvalue of the Hessian matrix $\nabla^2 f(\bar{x})$. Thus, for all $\|d\|$ sufficiently small: $f(\bar{x} + d) - f(\bar{x}) \geq \frac{a}{2}\|d\|^2 > 0$ which implies that $\bar{x}$ is a strict local minimum along d $\forall d \in \mathcal{B}_\varepsilon(0)$. ■

Since this holds $\forall d \in \mathbb{R}^n$ such that $\bar{x} + d \in S \cap \mathcal{B}_\epsilon(\bar{x})$, f is locally strictly convex.

If a point satisfies this (sufficient) condition then we've sure that it's a local optimum (not a saddle point)

How these results can be **strengthen** if the problem that we optimize is **convex**?

Convex problems

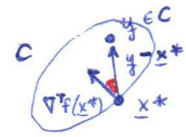
$$\min_{x \in C \subseteq \mathbb{R}^n} f(x) \quad \text{where } C \subseteq \mathbb{R}^n \text{ convex and } f: C \to \mathbb{R} \text{ convex}$$

For convex problems every local minimum is a global minimum.

Necessary and sufficient (NS) conditions: (global optimality)

Let f be convex and C^1 on $C \subseteq \mathbb{R}^n$ convex. x^* is a *global minimum* of f on C if and only if

$$\nabla^t f(x^*)(y - x^*) \geq 0 \qquad \forall y \in C.$$



$\blacktriangledown \leq 180°$

Same as the first order necessary condition for local optimality with the difference that here we don't have $\forall d$ feasible but $y - x^* \quad \forall y \in C$.

Proof:

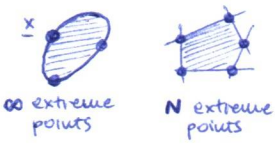
N. conditions: if $f \in C^1$ and x^* is a local minimum (also global minimum due to convexity) then

$\nabla^t f(x^*) d \geq 0 \quad \forall d$ feasible directions at x^*, namely $\forall d = y - x^*$ with $y \in C$.

S. conditions: f is convex if and only if

$$f(y) \geq f(x^*) + \nabla^t f(x^*)(y - x^*) \qquad \forall y \in C.$$

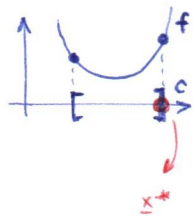
Then $\nabla f(x^*)(y - x^*) \geq 0$ implies that $f(y) \geq f(x^*)$ for every $y \in C$. □



Recall: Let $C \subseteq R^n$ be convex. Then $x \in C$ is an *extreme point* of C if it cannot be expressed as a convex combination of two different points of C.

Property: (maximization of convex functions)

Let f be a convex function defined on a convex bounded closed set C. If f has a (finite) *maximum* over C, then $\exists$ an *optimal extreme point* of C.



If the problem is not unbounded, if we maximize a convex function over a convex set then the maximum is reached in an extreme point ⟹ we just need to focus on extreme points so either:

C ∂C / N points

Proof:

Suppose that x^* is a global maximum of f over C, but not an extreme point. ($x^* \in \text{int}(C)$)

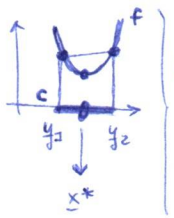
1) Verify that the maximum is achieved at a point on the boundary ∂C.

Since C is convex bounded and closed, for any $x^* \in int(C)$ there exist $y_1, y_2 \in \partial C$ and $\alpha \in [0,1]$ such that $x^* = \alpha y_1 + (1-\alpha) y_2$.

Due to convexity of f, we have

$$f(x^*) \leq \alpha f(y_1) + (1-\alpha) f(y_2) \leq \min\{f(y_1), f(y_2)\}.$$

Thus also y_1 and y_2 are global maxima.



⟹ from now we can assume (w.l.o.g.) that the global optimum is on the boundary

2) Suppose $x^* \in \partial C$ is not an extreme point.

Consider $T_1 = C \cap H$, where H is a supporting hyperplane at $x^* \in \partial C$.

Clearly $dim(T_1) \leq n - 1$.

Since T_1 is compact, $\exists$ a global optimum x_1 of f over T_1 such that

$$\max_{x \in T_1} f(x) = f(x_1) = f(x^*)$$

and, as previously, we have $x_1 \in \partial T_1$.

Claim: If x_1 is an extreme point of T_1, x_1 is also an extreme point of C.

If x_1 is not an extreme point of T_1, we similarly define $T_2, \ldots$

In the worst case $dim(T_n) = 0$. Such an isolated x_n is clearly an extreme point. Since an extreme point of T_i is also an extreme point of T_{i-1}, x_n must be an extreme point of C. □

since $x^* \in \partial C$ and the set is closed and convex then $\exists H$ separation hyperplane ("supporting hyperplane") which supports C at x^*:

$T_1 := H \cap C$

Application:

if f is linear (concave and convex) and we maximize it on a bounded closed set then we just consider the extreme points:

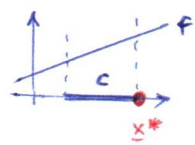


Illustration:

Special case: Linear programming

4.3 Iterative methods and convergence

Generic Nonlinear Optimization (NO) problem:

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_i(x) \leq 0 \quad 1 \leq i \leq m \\ & x \in S \subseteq \mathbb{R}^n \end{aligned}$$

← m non-linear constraints

If $X = \{x \in S : g_i(x) \leq 0,\ 1 \leq i \leq m\} \subset \mathbb{R}^n$ then constrained problem.

Difficulty depends on f and X. Usually f and g_i are at least continuously differentiable.

In some cases (e.g., LP and combinatorial optimization) an optimal solution can be found in a finite number of elementary operations.

Efficiency depends on how this number grows with the instance size (polynomial vs exponential).

Most NO methods are **iterative**

- start from $\underline{x}_0 \in X$
- generate (based on preceding $\underline{x}_k$, f and its derivatives) a sequence $\{\underline{x}_k\}_{k\geq 0}$ that "converges" to a point of $\Omega = \{$"desired solutions"$\}$.

Different meanings of "**converge**" and "**desired solutions**":

- $\{\underline{x}_k\}_{k\geq 0}$ converges to a point of Ω,
  $\exists$ a limit point of $\{\underline{x}_k\}_{k\geq 0}$ which belongs to Ω (a good estimate after a sufficiently large number of iterations)
- Ω = set of global solutions, (global optima)
  Ω = set of candidate points which satisfy 1st/2nd order necessary optimality conditions (e.g. $\Omega = \{\underline{x} \in \mathbb{R}^n : \nabla f(\underline{x}) = \underline{0}\}$ if $X = \mathbb{R}^n$).

Often but not always descent methods: $f(\underline{x}_{k+1}) < f(\underline{x}_k)$ for each k

Interested in robust and efficient methods.

1) **Robustness** associated to global convergence (≠ reaching the global optimal)

Definition: An algorithm is **globally (locally) convergent** if $\{\underline{x}_k\}_{k\geq 0}$ satisfies one of previous properties for any $\underline{x}_0 \in X$ (only for $\underline{x}_0$ in a neighborhood of an $\underline{x}^* \in \Omega$).

neighborhood of a desired solution (which is not necessarily an optima)

2) **Efficiency** characterized by convergence speed

Assume that $\lim_{k\to\infty} \underline{x}_k = \underline{x}^*$ where $\underline{x}^* \in \Omega$

Definitions: $\{\underline{x}_k\}_{k\geq 0}$ *converges* to $\underline{x}^*$ with *order* $p \geq 1$ if $\exists\ r > 0$ and $k_0 \in \mathbb{N}$ such that

$$\|\underline{x}_{k+1} - \underline{x}^*\| \leq r \|\underline{x}_k - \underline{x}^*\|^p \qquad \forall k \geq k_0.$$

Largest p is the **order of convergence** and smallest $r > 0$ is the **rate**.

If $p = 1$ and $r < 1$ **linear** convergence, if $p = 1$ and $r \geq 1$ **sublinear** convergence.

N.B.: If $p = 1$ the distance w.r.t. $\underline{x}^*$ decreases at each iteration by a factor r.

we want methods that provide desired solutions independ. on how far we are from the desired solution when we start

Example: $1 + \frac{1}{k} \to 1$ with $r = 1$ and $1 + \frac{1}{2^k} \to 1$ with $r = \frac{1}{2}$

$$\|x_{k+1} - x^*\| \leq r \|x_k - x^*\|^p \qquad \forall k \geq k_0$$

$$\lim_{k\to\infty} \frac{\left|1 + \frac{1}{k+1} - 1\right|}{\left|1 + \frac{1}{k} - 1\right|^p} = \left[\frac{k^p}{k+1}\right] \simeq \begin{cases} 0 & p < 1 \\ 1 & p = 1 \\ \infty & p > 1 \end{cases} \Longrightarrow \boxed{p = 1,\ r = 1}$$ sublinear convergence (p = largest p s.t. the rate is finite)

$$\lim_{k\to\infty} \frac{\left|1 + \frac{1}{2^{k+1}} - 1\right|}{\left|1 + \frac{1}{2^k} - 1\right|^p} = \left[\frac{\frac{1}{2}\left(2^{p-1}\right)^k}{}\right] = \begin{cases} \vdots \end{cases} \Longrightarrow \boxed{p = 1,\ r = \frac{1}{2}}$$ linear convergence

Definition: The **convergence is superlinear** if there exists $\{r_k\}_{k\geq 0}$ with $\lim_{k\to\infty} r_k = 0$ such that

$$\|\underline{x}_{k+1} - \underline{x}^*\| \leq r_k \|\underline{x}_k - \underline{x}^*\| \qquad \forall k \geq k_0.$$ (p=1 but the rate is not constant)

Example: $1 + \frac{1}{k^k}$

Definition: If $p = 2$ (and r not necessarily < 1), the **convergence is quadratic**.

Example: $1 + \frac{1}{2^{2^k}}$

4.4 Line search methods

Unconstrained optimization problem:

$$\min_{\underline{x}\in\mathbb{R}^n} f(\underline{x})$$ → no space constraints

with $f : \mathbb{R}^n \to \mathbb{R}$ of class $\mathcal{C}^1$ or $\mathcal{C}^2$ and bounded below.

Iterative methods start from $\underline{x}_0 \in \mathbb{R}^n$ and generate $\{\underline{x}_k\}_{k\geq 0}$ "converging" to a point of the set Ω of the "desired solutions".

accuracy parameter (related to the stopping criterium)

1) General scheme

Select $\underline{x}_0$ and $\varepsilon > 0$, set $k := 0$

Repeat

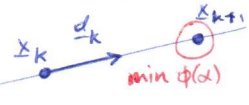
- Choose a *search direction* $\underline{d}_k \in \mathbb{R}^n$
- Determine *step length* $\alpha_k > 0$ along $\underline{d}_k$ such that $\min_{\alpha \geq 0} \phi(\alpha) = f(\underline{x}_k + \alpha \underline{d}_k)$
- Set $\underline{x}_{k+1} := \underline{x}_k + \alpha_k \underline{d}_k$ and $k := k + 1$

Until termination criterion is satisfied

Termination criterion: $\|\nabla f(\underline{x}_k)\| < \varepsilon$ or $|f(\underline{x}_k) - f(\underline{x}_{k+1})| < \varepsilon$ or $\|\underline{x}_{k+1} - \underline{x}_k\| < \varepsilon$

Often α_k is determined in an approximate way so that $f(\underline{x}_{k+1}) < f(\underline{x}_k)\ \forall k \geq 0$.

Flexibility in the choice of $\underline{d}_k$ and α_k, the method efficiency depends on both!



we look for a direction $\underline{d}_k$ and we try to optimize the objective function f along the half-line going from $\underline{x}_k$ along $\underline{d}_k$. We can see this function f restricted on the direction $\underline{d}_k$ as a function $\phi(\alpha) = f(\underline{x}_k + \alpha \underline{d}_k)$.

In this method we're optimizing on a line (that's why "line search mthd.") so this is a 1D optimization problem.

2) Search directions

In many line search methods, i.e., iterative methods based on search directions,

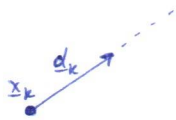
$$\underline{d}_k = -D_k \nabla f(\underline{x}_k)$$

with a positive definite $n \times n$ matrix D_k.

Thus $\underline{d}_k$ is a *descent direction* because

$$\nabla^t f(\underline{x}_k) \underline{d}_k = -\nabla^t f(\underline{x}_k) D_k \nabla f(\underline{x}_k) < 0.$$

we are guaranteed that if we move from $\underline{x}_k$ along the direction $\underline{d}_k$ then the objective function value will decrease



Example 1: Gradient method

we have to be able to compute at least the gradient

Given $f \in C^1$, consider linear approximation of $f(\underline{x}_k + \underline{d})$ at $\underline{x}_k$

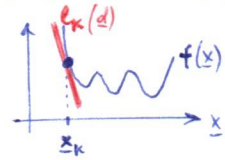
$$l_k(\underline{d}) := f(\underline{x}_k) + \nabla^t f(\underline{x}_k) \underline{d}$$ ← linear approximation of the function:

and choose $d_k \in \mathbb{R}^n$ minimizing $l_k(\underline{d})$ on sphere of radius $\|\nabla f(\underline{x}_k)\|$:

$$\begin{array}{ll} \min & \nabla^t f(\underline{x}_k) \underline{d} \\ \text{s.t.} & \|\underline{d}\| = \|\nabla f(\underline{x}_k)\|. \end{array} \qquad (1)$$

we neglect $f(\underline{x}_k)$ since it's a constant

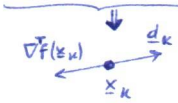
(*)



Since $\nabla^t f(\underline{x}_k) \underline{d} = \|\nabla f(\underline{x}_k)\| \|\underline{d}\| \cos(\theta)$, (1) is minimized when $\cos(\theta) = -1$, namely $\theta = \pi$.

Steepest descent direction:

$$\underline{d}_k = -\nabla f(\underline{x}_k)$$



where $D_k = I_n$.

Clearly $\underline{d}_k$ is a descent direction if $\nabla f(\underline{x}_k) \neq \underline{0}$.

$\Longrightarrow$ $\underline{x}_k$ is a non stationary pt.

Idea of the method: we consider the linear approx. around $\underline{x}_k$ and we minimize this. If we minimize it over all $\mathbb{R}^n$ we get an unbounded optimization problem, so we have to minimize it over a bounded region → A BALL. (*)

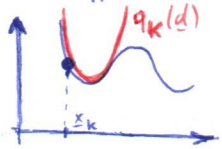
Example 2: Newton method

Given $f \in C^2$ and $H(\underline{x}_k) = \nabla^2 f(\underline{x}_k)$. → The **idea** is to extract more information from the objective function than just the linear approximation: we approximate f with a quadratic approximation at $f(\underline{x}_k)$

Consider quadratic approximation of $f(\underline{x}_k + \underline{d})$ at $\underline{x}_k$

$$q_k(\underline{d}) := f(\underline{x}_k) + \nabla^t f(\underline{x}_k) \underline{d} + \frac{1}{2} \underline{d}^t H(\underline{x}_k) \underline{d}$$

and choose $d_k \in \mathbb{R}^n$ and α_k leading to a stationary point of $q_k(\underline{d})$. ($\alpha_k = 1$)



this approximation "hugs" the graph a little more closely than the linear approximation

Since $\nabla_{\underline{d}}\, q_k(\underline{d}) = \underline{0}$ implies $\nabla^t f(\underline{x}_k) + \underline{d}^t H(\underline{x}_k) = \underline{0}$, if $H^{-1}(\underline{x}_k)$ exists:

Newton direction is:

$$\underline{d}_k = -H^{-1}(\underline{x}_k) \nabla f(\underline{x}_k),$$

where $D_k = H^{-1}(\underline{x}_k)$.

positive definite

If $H(\underline{x}_k)$ is p.d. and $\nabla f(\underline{x}_k) \neq \underline{0}$, $\underline{d}_k$ is a descent direction

$$\nabla^t f(\underline{x}_k) \underline{d}_k = -\nabla^t f(\underline{x}_k) H^{-1}(\underline{x}_k) \nabla f(\underline{x}_k) \leq -\sigma_k \|\nabla f(\underline{x}_k)\|^2 < 0$$

for a $\sigma_k > 0$.

If $H(\underline{x}_k)$ is not p.d., $\underline{d}_k$ may not be defined ($\nexists\ H^{-1}(\underline{x}_k)$) or may be an ascent direction.

3) Step length

To guarantee global convergence it suffices to determine an approximate solution α_k of *line search*:

$$\min_{\alpha \geq 0} \phi(\alpha) = f(\underline{x}_k + \alpha \underline{d}_k).$$

Different methods generate α_k and stop when appropriate conditions are satisfied (simple, after a few iterations).

we never stop at the real optimal α (it's not worthy)

(If $\underline{x}_0$ is very bad we may have a bad direction $\underline{d}$, we don't want to spend too much time on a bad direction)

$f(\underline{x}_k + \alpha_k \underline{d}_k) < f(\underline{x}_k)$ does not suffice.

Basic principles:

- α must not be too small (to avoid premature convergence)
- α must not be too large (to avoid oscillations)

Wolfe conditions:

Sufficient reduction:

$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0) \quad \text{con } c_1 \in [0, 1]$$

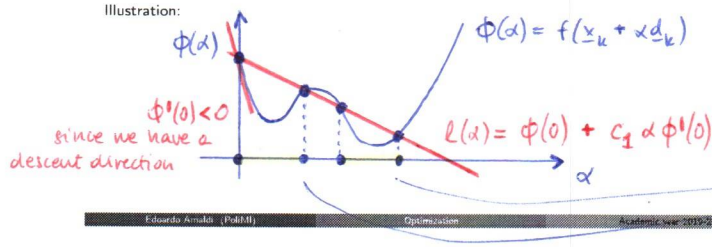
$:= \ell(\alpha)$

which is equivalent to

$$f(\underline{x}_k + \alpha \underline{d}_k) \leq f(\underline{x}_k) + c_1 \alpha \nabla^t f(\underline{x}_k) \underline{d}_k \qquad \text{(Armijo criterion)}$$

$\phi'(0) < 0$ since $\underline{d}_k$ is a descent direction, $c_1 \leq 1/2$ so that it is satisfied by the minimum of a quadratic convex $\phi(\alpha)$ (see exercise 6.5).

Illustration:



acceptable values of α $= \{\alpha : \phi(\alpha) \leq \ell(\alpha)\}$

Note: if we take $c_1 \leq \frac{1}{2}$ then if the function is 1D is quadratic convex function $\Rightarrow$ among the acceptable values of α there will be $\bar{\alpha}$ for which $\phi(\bar{\alpha})$ is the global minimum (of $\phi(\cdot)$)

To avoid too small steps also condition:

$$\phi'(\alpha) \geq c_2 \phi'(0) \quad \text{con } c_2 \in (c_1, 1)$$

which is equivalent to

$$\nabla^t f(\underline{x}_k + \alpha \underline{d}_k) \underline{d}_k \geq c_2 \nabla^t f(\underline{x}_k) \underline{d}_k.$$

Problem: among those α's some of them are too small so we need a second condition which guarantee α to be sufficiently large

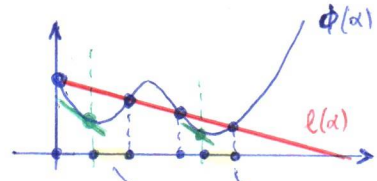
In general $c_2 = 0.9$ for (quasi)-Newton and $c_2 = 0.1$ for non-linear conjugate gradient.

Weak Wolfe conditions:

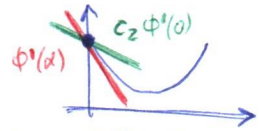
$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0) \qquad (2)$$
$$\phi'(\alpha) \geq c_2 \phi'(0) \qquad (3)$$

with $0 < c_1 < c_2 < 1$.

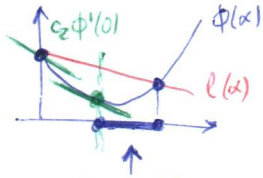
Illustration:



acceptable values of α (smaller regions than before)



we consider the slope corresponding to $c_2 \phi'(0)$ (which is less negative than $\phi'(0)$) and we consider only the α which have a slope that is less negative than $c_2 \phi'(0)$:



we consider only these α's

Strong Wolfe conditions:

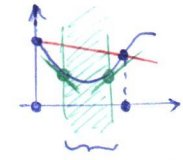
$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0) \qquad (4)$$
$$|\phi'(\alpha)| \leq c_2 |\phi'(0)| \qquad (5)$$

with $0 < c_1 < c_2 < 1$.

Exclude values of α with $\phi'(\alpha)$ too positive, far from stationary points of ϕ.

Conditions are invariant w.r.t. affine transformation of the variables.

Actually we should exclude even the α's for which the slope is too positive (not only the α's for which the slope is too negative). Why? If the slope is too positive we're close to a stationary point: ( )



acceptable α's

Do these regions of α's always exist? (Y) Are they difficult to be found? (N):

Proposition:

If $f : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^1$ and $\underline{d}_k$ descent direction at $\underline{x}_k$ such that f is bounded below along $\{\underline{x}_k + \alpha \underline{d}_k : \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$ there exist intervals of step lengths satisfying the Wolfe conditions (weak and strong).

Simple consequence of the mean value theorem.

How can we find those α's?

Metods for 1-D search

Many methods (with/without derivatives) to determine an approximate solution α_k of

$$\min_{\alpha \geq 0} \phi(\alpha) = f(\underline{x}_k + \alpha \underline{d}_k)$$

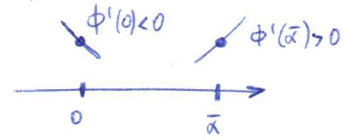
satisfying appropriate conditions (e.g. Wolfe) which guarantee global convergence.

In general, two phases:

- determine $[\alpha_{min}, \alpha_{max}]$ containing "acceptable" step lengths ("bracketing phase"),
- select a good value α within $[\alpha_{min}, \alpha_{max}]$ via bisection or interpolation.

Bisection

$\phi \in \mathcal{C}^1$, $\phi'(0) < 0$ since $\underline{d}_k$ descent direction and $\exists\ \overline{\alpha}$ such that $\phi'(\alpha) > 0$ for $\alpha \geq \overline{\alpha}$. :

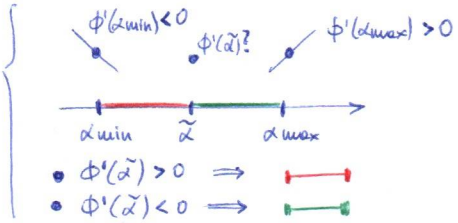


Start from $[\alpha_{min}, \alpha_{max}]$ with $\phi'(\alpha_{min}) < 0$ and $\phi'(\alpha_{max}) > 0$ and iteratively reduce it.

Iteration: Set $\tilde{\alpha} = \frac{1}{2}(\alpha_{min} + \alpha_{max})$

IF $\phi'(\tilde{\alpha}) > 0$ THEN $\alpha_{max} := \tilde{\alpha}$

IF $\phi'(\tilde{\alpha}) < 0$ THEN $\alpha_{min} := \tilde{\alpha}$



Linear convergence with rate 1/2

To find initial $[\alpha_{min}, \alpha_{max}]$:

1) $\alpha_{min} := 0$ e $s := s_0$ (given step)

2) Compute $\phi'(s)$

IF $\phi'(s) < 0$ THEN $\alpha_{min} := s$, $s := 2s$ and GOTO 2)

IF $\phi'(s) > 0$ THEN $\alpha_{max} := s$ and STOP

How can we adapt the Bisection method to find α satisfing the Wolfe conditions?

Principle can be adapted to determine α_k satisfying Wolfe conditions.

Wolfe (2): $\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0)$

Wolfe (3): $\phi'(\alpha) \geq c_2 \phi'(0)$

Procedure:

i) Select $\alpha > 0$ and set $\alpha_{min} = \alpha_{max} = 0$

ii) IF α satisfies Wolfe (2) THEN GOTO iii)

ELSE $\alpha_{max} := \alpha$, $\alpha := \frac{\alpha_{min} + \alpha_{max}}{2}$ and GOTO ii)

iii) IF α satisfies Wolfe (3) THEN $\alpha_k = \alpha$ and STOP

ELSE $\alpha_{min} := \alpha$

$$\alpha := \begin{cases} 2\alpha_{min} & \text{if } \alpha_{max} = 0 \\ \frac{1}{2}(\alpha_{min} + \alpha_{max}) & \text{if } \alpha_{max} > 0 \end{cases}$$

and GOTO ii)

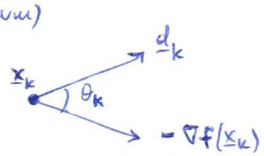
Proposition: If $f \in \mathcal{C}^1$ is bounded below along ray $\{\underline{x}_k + \alpha \underline{d}_k : \alpha \geq 0\}$, the procedure stops after a finite number of iterations and yields α_k satisfying Wolfe conditions.

4) Global convergence of line search methods

Suitable assumptions on step lengths α_k and directions $\underline{d}_k$ can guarantee global convergence. (= $\forall x_0$ we have convergence to the desired solution ≠ convergence to the global optimum)

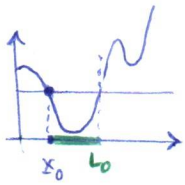
Key aspect: angle θ_k between $\underline{d}_k$ and $-\nabla f(\underline{x}_k)$

$$\cos(\theta_k) = -\frac{\nabla^t f(\underline{x}_k)\underline{d}_k}{\|\nabla f(\underline{x}_k)\| \|\underline{d}_k\|}$$



General result showing how far $\underline{d}_k$ can deviate from $-\nabla f(\underline{x}_k)$ and still give rise to globally convergent iterations.

For a proof assuming weak Wolfe conditions, see J. Noceal, S. Wright, Numerical Optimization, Springer 1999, p. 43-44.



x_0 = initial solution generated by the line search method

$\min_{x \in \mathbb{R}^n} f(x)$

Theorem: (Zoutendijk)

Consider any line search method iteration with descent direction d_k and α_k satisfying Wolfe conditions. Suppose f is bounded below on $\mathbb{R}^n$, $f \in \mathcal{C}^1$ on open set N containing $L_0 = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$ and $\nabla f(x)$ is Lipschitz continuous on N, that is, $\exists\, L > 0$ such that

$$\|\nabla f(x) - \nabla f(\bar{x})\| \le L\|x - \bar{x}\| \qquad \forall x, \bar{x} \in N.$$

Then

$$\sum_{k \ge 0} \cos^2(\theta_k)\|\nabla f(x_k)\|^2 < +\infty. \tag{6}$$

Condition (6) implies $\cos^2(\theta_k)\|\nabla f(x_k)\|^2 \to 0$ when $k \to \infty$.

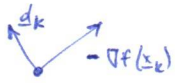
If $\cos\theta_k \ge \delta > 0 \quad \forall k \ge 0$ then (6) implies that $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ for any x_0. = $\{x_k\}_{k\ge 0}$ generated by any line search method satisfying those conditions converges to a stationary point

According to (6), $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ provided that the directions d_k are never too close to orthogonality with $-\nabla f(x_k)$. (since $\cos(\theta_k) > 0$)

so we're avoiding ⊥ directions.

Consequence: The gradient method ($\cos\theta_k = 1$) satisfying Wolfe conditions is globally convergent.

which is nice. If we had:



then along d_k we would not decrease the objective function.

If D_k symmetric and p.d. $\forall k \ge 0$ and $\exists$ constant M such that

$$\|D_k\|\|D_k^{-1}\| \le M \quad \forall k \ge 0$$

(bounded condition number), it can be verified that

$$\cos\theta_k \ge 1/M.$$

In such cases Newton and quasi-Newton methods are globally convergent.

4.5 Gradient method

- very simple
- very light computationally
- globally convergent (but the convergence can be very slow)

Given $f : \mathbb{R}^n \to \mathbb{R}$ with $f \in \mathcal{C}^1$, look for a stationary point.

Gradient method with exact 1-D search:

Choose x_0, set $k := 0$

Iteration k:

$d_k := -\nabla f(x_k)$

Determine $\alpha_k > 0$ such that $\min_{\alpha \ge 0} \phi(\alpha) = f(x_k + \alpha d_k)$

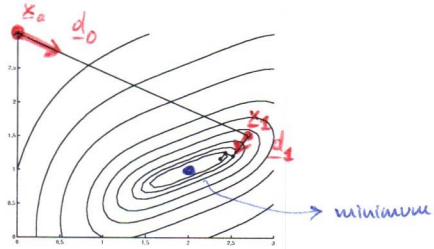
$x_{k+1} := x_k + \alpha_k d_k$ — we solve it exactly (not approximately)

$k := k + 1$

Termination criteria: $\|\nabla f(x_k)\| < \varepsilon$ or $|f(x_k) - f(x_{k+1})| < \varepsilon$ or $\|x_{k+1} - x_k\| < \varepsilon$.

Property: If 1-D search is exact, the successive directions are orthogonal.

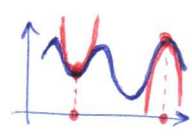
Since α_k such that $\min_{\alpha \ge 0} \phi(\alpha) = f(x_k + \alpha d_k)$,
$\phi'(\alpha_k) = \nabla^t f(x_k + \alpha_k d_k) d_k = \nabla^t f(x_{k+1}) d_k = 0$
and hence $d_{k+1}^t d_k = -\nabla^t f(x_{k+1}) d_k = 0$.



Example: zig-zag trajectory, very slow convergence → the fact that successive directions are orthogonal is not so good

First we consider the case of quadratic strictly convex functions.

Important since any $\mathcal{C}^2$ function can be well approximated around any local/global minimum by such a function.



⇒ So when we discuss the method we discuss it first for the quadratic strictly convex functions and then we'll extend the method for arbitrary functions

Quadratic strictly convex functions:

$$f(\underline{x}) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{b}^t \underline{x} \quad \text{with } Q \text{ symmetric and p.d.}$$

positive definite → exact 1D search

since it's convex then a local minimum is a global minimum

Global minimum is unique solution of $\nabla f(\underline{x}) = Q\underline{x} - \underline{b} = \underline{0}$ (hence of $Q\underline{x} = \underline{b}$) and α_k can be determined explicitly:

$$\phi(\alpha) = f(\underline{x}_k - \alpha \nabla f(\underline{x}_k)) = \frac{1}{2}(\underline{x}_k - \alpha \nabla f(\underline{x}_k))^t Q(\underline{x}_k - \alpha \nabla f(\underline{x}_k)) - \underline{b}^t(\underline{x}_k - \alpha \nabla f(\underline{x}_k))$$

$$\phi'(\alpha) = -\nabla^t f(\underline{x}_k) Q(\underline{x}_k - \alpha \nabla f(\underline{x}_k)) + \underline{b}^t \nabla f(\underline{x}_k) = 0$$

Since $\nabla^t_{\underline{x}} f(\underline{x}_k) = \underline{x}_k^t Q - \underline{b}^t$ implies that $\underline{b}^t = -\nabla^t f(\underline{x}_k) + \underline{x}_k^t Q$,

$$-\nabla^t f(\underline{x}_k) Q \underline{x}_k + \alpha \nabla^t f(\underline{x}_k) Q \nabla f(\underline{x}_k) + (-\nabla^t f(\underline{x}_k) + \underline{x}_k^t Q)\nabla f(\underline{x}_k) = 0$$

thus

$$\alpha_k = \frac{\nabla^t f(\underline{x}_k)\nabla f(\underline{x}_k)}{\nabla^t f(\underline{x}_k) Q \nabla f(\underline{x}_k)}$$

and in terms of $\underline{d}_k$:

$$\alpha_k = \frac{\underline{d}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}.$$

(assumptions: $\nabla f(\underline{x}^*) = 0$)

Convergence analysis

Often consider convergence rate of $f(\underline{x}_k) \to f(\underline{x}^*)$ instead of $\|\underline{x}_k - \underline{x}^*\| \to 0$ when $k \to \infty$.

($\{\underline{x}_k\} \xrightarrow{k} \underline{x}^*$)

Proposition: If $H(\underline{x}^*)$ is p.d., $\underline{x}_k$ converges (super)linearly at $\underline{x}^*$ w.r.t. $|f(\underline{x}_k) - f(\underline{x}^*)|$ if and only if it converges in the same way w.r.t. $\|\underline{x}_k - \underline{x}^*\|$.

= looking at the speed of convergence of $|f(\underline{x}_k) - f(\underline{x}^*)| \to 0$ or $\|\underline{x}_k - \underline{x}^*\| \to 0$ is the same thing

⇓

So very often we'll consider the convergence: $f(\underline{x}_k) \to f(\underline{x}^*)$

Indeed

$$f(\underline{x}) \approx f(\underline{x}^*) + \frac{1}{2}(\underline{x} - \underline{x}^*)^t H(\underline{x}^*)(\underline{x} - \underline{x}^*)$$

and $\exists$ a neighborhood $N(\underline{x}^*)$ such that

$$\lambda_1' \|\underline{x} - \underline{x}^*\|^2 \le |f(\underline{x}) - f(\underline{x}^*)| \le \lambda_n' \|\underline{x} - \underline{x}^*\|^2 \qquad \forall \underline{x} \in N(\underline{x}^*)$$

with $\lambda_1' = \lambda_1 - \varepsilon > 0$ and $\lambda_n' = \lambda_n + \varepsilon$, where $\varepsilon > 0$ and $0 < \lambda_1 \le \ldots \le \lambda_n$ are the eigenvalues of $H(\underline{x}^*)$.

N.B.: This equivalence does not hold in general (e.g., functions non everywhere $\mathcal{C}^1$)

example:

Quadratic strictly convex functions:

$f(\underline{x}) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{b}^t \underline{x}$ and weighted norm $\|\underline{x}\|_Q^2 := \underline{x}^t Q \underline{x}$.

Since $Q\underline{x}^* = \underline{b}$,

$\frac{1}{2}\|\underline{x} - \underline{x}^*\|_Q^2 = \frac{1}{2}(\underline{x} - \underline{x}^*)^t Q(\underline{x} - \underline{x}^*) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{x}^{*t} Q \underline{x} + \frac{1}{2}\underline{x}^{*t} Q \underline{x}^* = \ldots = f(\underline{x}) - f(\underline{x}^*)$.

Theorem: If gradient method with exact 1-D search is applied to any quadratic strictly convex $f \in \mathcal{C}^2$, for any $\underline{x}_0$ we have $\lim_{k\to\infty} \underline{x}_k = \underline{x}^*$ and

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

linear convergence of: $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$

this means that the method is globally convergent

where $0 < \lambda_1 \le \ldots \le \lambda_n$ are the eigenvalues of Q.

r^2: notice that if the spectrum of Q is small (λ_1 and λ_n are close) then the rate r will be small. The smaller the spectrum the smaller the convergence, and so, the faster the convergence

Proof sketch:

Zoutendijk's theorem implies global convergence.

Since exact 1-D search, easy to verify that

$$\|\underline{x}_{k+1} - \underline{x}^*\|_Q^2 = \left(1 - \frac{\underline{g}_k^t \underline{g}_k}{(\underline{g}_k^t Q \underline{g}_k)(\underline{g}_k^t Q^{-1} \underline{g}_k)}\right) \|\underline{x}_k - \underline{x}^*\|_Q^2,$$

where $\underline{g}_k = Q\underline{x}_k - \underline{b}$. $= \nabla f(\underline{x}_k)$

Then just apply Kantorovich inequality:

If Q p.d. (with λ_1 and λ_n smallest and largest eigenvalues), for each $\underline{x} \neq \underline{0}$ we have

$$\frac{(\underline{x}^t \underline{x})^2}{(\underline{x}^t Q \underline{x})(\underline{x}^t Q^{-1} \underline{x})} \ge \frac{4\lambda_n \lambda_1}{(\lambda_n + \lambda_1)^2}.$$

□

If $\lambda_1 = \lambda_n$ ($Q = \gamma I$), method "converges" in one iteration.

Upper bound (1) is reached for some choices of $\underline{x}_0$ (Aikake). ⟹ depending on $\underline{x}_0$ the gradient method can be very slow

Linear convergence whose rate depends on *condition number* $\kappa = \frac{\lambda_n}{\lambda_1}$ of Q:

$$r = \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right) = \left(\frac{\kappa - 1}{\kappa + 1}\right)$$

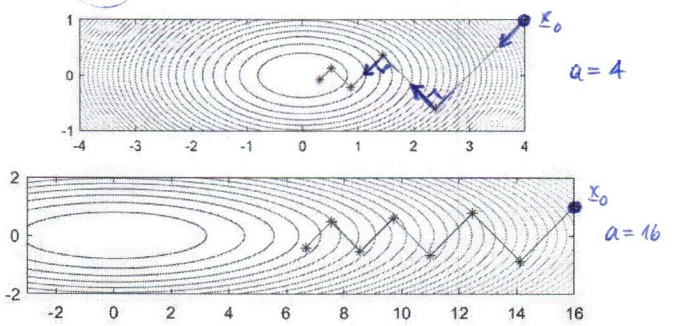
the closer κ to 1 the smaller r; if the spektrum of Q is very wide then $\kappa \gg 1$ and $r \approx 1$.

⟹ sublinear convergence

quadratic strictly convex function

Example:

$\min f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{a}{2}x_2^2$ with $a \geq 1$ and hence eigenvalues $\frac{1}{2}$ and $\frac{a}{2}$



Some points of the sequence $\{\underline{x}_k\}$ for $a = 4$ (top) and $a = 16$ (bottom), starting from $\underline{x}_0 = \begin{pmatrix} a \\ 1 \end{pmatrix}$.

Arbitrary non linear functions:

Theorem: If $f \in \mathcal{C}^2$ and gradient method with exact 1-D search converges to $\underline{x}^*$ with $H(\underline{x}^*)$ p.d., then

$$f(\underline{x}_{k+1}) - f(\underline{x}^*) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 [f(\underline{x}_k) - f(\underline{x}^*)]$$

convergence rate in terms of $f(\underline{x}_k)$ and not $\underline{x}_k$

where $0 < \lambda_1 \leq \ldots \leq \lambda_n$ are eigenvalues of $H(\underline{x}^*)$.

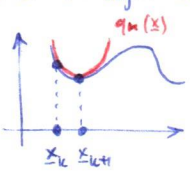
We cannot expect better convergence with inexact (approximate) 1-D search.

α_k minimizing $\phi(\alpha)$ may not be the best choice, we could try to "extract" 2nd order information about $f(\underline{x})$.

Example: for $f(\underline{x})$ quadratic strictly convex, $\alpha_k = 1/\lambda_{k+1}$ lead to $\underline{x}^*$ in at most n iterations!

4.6 Newton method

it exploits more information than the gradient method (in fact we do a quadratic approximation of the obj. function)



Let $f \in \mathcal{C}^2$ and $H(\underline{x}) = \nabla^2 f(\underline{x})$.

Consider quadratic approximation of $f(\underline{x})$ at $\underline{x}_k$:

$$q_k(\underline{x}) := f(\underline{x}_k) + \nabla^t f(\underline{x}_k)(\underline{x} - \underline{x}_k) + \frac{1}{2}(\underline{x} - \underline{x}_k)^t H(\underline{x}_k)(\underline{x} - \underline{x}_k)$$

and choose as $\underline{x}_{k+1}$ a stationary point ($\nabla_{\underline{x}} q_k(\underline{x}) = \underline{0}$), namely

$$\nabla f(\underline{x}_k) + H(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k) = \underline{0}.$$

If $H(\underline{x}_k)$ is not singular, $H^{-1}(\underline{x}_k)$ exists and

$$\boxed{\underline{x}_{k+1} := \underline{x}_k - H^{-1}(\underline{x}_k)\nabla f(\underline{x}_k).}$$

If $H(\underline{x}_k)$ is p.d., $f \in \mathcal{C}^2$ implies that $H^{-1}(\underline{x}_k)$ p.d. over $N(\underline{x}_k)$ and iteration is well defined, otherwise $\underline{d}_k$ may not be a descent direction.

In the "pure" Newton method, $\alpha_k = 1$ for each k.

For f quadratic and strictly convex, global minimum in a single iteration.

Property: Newton method is invariant w.r.t. affine and non singular coordinate changes (see exercise 6.6).

! **Observation**: Newton method is not globally convergent, but very fast local convergence if $\underline{x}_0$ is sufficiently close to a desired solution.

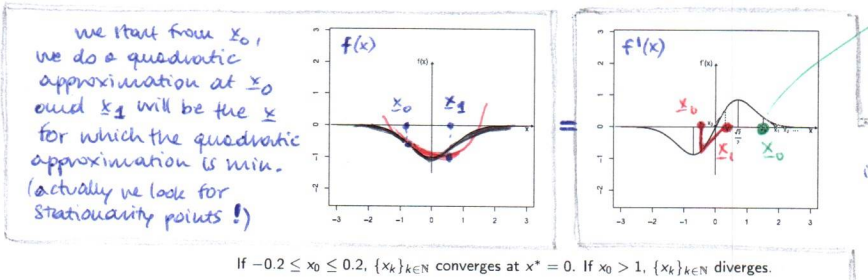
= it strongly depends on $\underline{x}_0$

if we start from this $\underline{x}_0$ then the sequence $\{\underline{x}_k\}_k$ is diverging

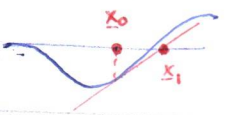
Example: $\min_{x \in \mathbb{R}} f(x) = -\exp(-x^2)$ with global minimum $x^* = 0$ and $f'(x) = 2x\exp(-x^2)$

finding $\min f(x)$ is equivalent to find the stationary point of $f'(x)$

we start from $\underline{x}_0$, we do a quadratic approximation at $\underline{x}_0$ and $\underline{x}_1$ will be the $\underline{x}$ for which the quadratic approximation is min. (actually we look for stationarity points !)



the quadratic approximation of $f(x)$ in terms of $f'(x)$ is a straight line. ⇒ same procedure as with $f(x)$ but with straight lines:



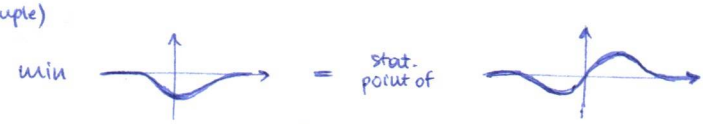
If $-0.2 \leq x_0 \leq 0.2$, $\{x_k\}_{k \in \mathbb{N}}$ converges at $x^* = 0$. If $x_0 > 1$, $\{x_k\}_{k \in \mathbb{N}}$ diverges.

Alternative interpretation of Newton method (1-D case): $\min f(\underline{x}) \iff f'(\underline{x}) = 0$

$f(x) \in C^2$ and look for x^* such that $f'(x) = 0$.

Method of tangents (Newton-Raphson) to determine the zeros of a 1-D function:

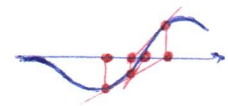
(previous slide example)



At iteration k, $f'(x)$ is approximated with the tangent at x_k

$$z = f'(x_k) + f''(x_k)(x - x_k)$$

x_{k+1} corresponds to the intersection with the x-axis: $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$



n-D case: Determine a stationary point of $f(\underline{x})$ by solving non linear system $\nabla f(\underline{x}) = \underline{0}$ with "Newton-Raphson" method.

(! locally convergent)

Theorem: (proof see Nocedal and Wright, Edition 1999, pages 52-53)
Suppose $f \in C^2$ and $\underline{x}^*$ such that $\nabla f(\underline{x}^*) = \underline{0}$ and $H(\underline{x}^*)$ p.d. and $\exists\ L > 0$ such that

$$\|H(\underline{x}) - H(\underline{y})\| \leq L\|\underline{x} - \underline{y}\| \qquad \forall \underline{x}, \underline{y} \in N(\underline{x}^*)$$

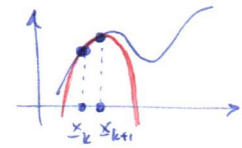
then, for $\underline{x}_0$ sufficiently close to local minimum $\underline{x}^*$,

i) $\{\underline{x}_k\} \to \underline{x}^*$ with a quadratic convergence order,
ii) $\{\|\nabla f(\underline{x}_k)\|\} \to 0$ quadratically when $k \to \infty$.

$\Longrightarrow$ local quadratic convergence

Disadvantages:

- If $H(\underline{x}_k)$ is singular the step is not defined.
- If $H^{-1}(\underline{x}_k)$ is not p.d., Newton direction may not be descent direction. $\rightarrow$
- Even for a descent direction $\alpha_k = 1$ may increase the value of f.
- Computation of $H^{-1}(\underline{x}_k)$ at each iteration ($O(n^3)$ complexity).
- Only *locally convergent*: if $\underline{x}_0$ is not close enough to $\underline{x}^*$, $\{\underline{x}_k\}_{k\geq 0}$ may not converge.
- Since $\{\underline{x}_k\}_{k\geq 0}$ converges from any $\underline{x}_0$ sufficiently close to any stationary point with non singular $\nabla^2 f(\underline{x})$, it may converge to local maxima.



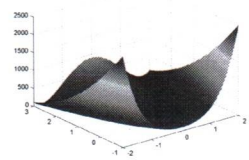
If we look for a stationary point of the quadratic approximation we get a point that increase the obj. function

For a comparison between gradient and Newton methods, see Nocedal and Wright, Numerical Optimization, Edition 1999, p. 199.

Rosenbrock function

$$f(\underline{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

quadratic and nonconvex.



Fourth computer laboratory: explore the considerable difference in convergence speed between various line search methods.

Modifications and extensions

1) If $\alpha_k = 1$ does not satisfy Wolfe (alternative) conditions then inexact 1-D search.

2) To guarantee global convergence

$$\underline{d}_k = -D_k \nabla f(\underline{x}_k)$$

with $D_k \neq [\nabla^2 f(\underline{x}_k)]^{-1}$. If D_k is symmetric and p.d., $\underline{d}_k$ is a descent direction.

trade off between the gradient method direction and the Newton direction: this $\underline{d}_k$ is the linear combination of the two matrices D_k for the two methods

Trade-off between steepest descent and Newton directions:

$$D_k := (\varepsilon_k I + \nabla^2 f(\underline{x}_k))^{-1}$$

where $\varepsilon_k > 0$ are smallest values such that eigenvalues of $(\varepsilon_k I + \nabla^2 f(\underline{x}_k))$ are $\geq \delta > 0$. Such ε_k making D_k p.d. always exist.

Coincides with "pure" Newton method when getting closer to a local minimum.

4.7 Conjugate direction methods

Aim: faster convergence than gradient method and lower computational load than Newton method.

First consider quadratic strictly convex functions

$$q(\underline{x}) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{b}^t \underline{x}$$

with Q $n \times n$ symmetric and p.d.

Definition: Given $n \times n$ and symmetric Q, two nonzero $\underline{d}_1, \underline{d}_2 \in \mathbb{R}^n$ are **Q-conjugate** if $\underline{d}_1^t Q \underline{d}_2 = 0$.

If $Q = I$, usual notion of orthogonality.

Example: $\min \; -12x_2 - 4x_1^2 + 4x_2^2 + 4x_1x_2$

$\Longrightarrow$ Hessian matrix Q: $Q = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$, $\underline{d}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\underline{d}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ are Q-conjugate : $\underline{d}_1^T Q \underline{d}_2 = 0$ (example in PDF)

(pairwise Q-conjugate)

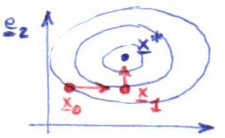
Proposition: If nonzero $\underline{d}_0, \ldots, \underline{d}_k$ are mutually Q-conjugate w.r.t. Q p.d., then $\underline{d}_0, \ldots, \underline{d}_k$ are linearly independent.

Proof: Suppose $\sum_{i=1}^{k} \lambda_i \underline{d}_i = \underline{0}$. We multiply by $Q\underline{d}_j$: $Q\underline{d}_j$ for j : $1 \le j \le k$, $j \neq i$. Now $\underline{d}_i^T Q \underline{d}_j = 0 \quad \forall i \neq j$ (since they're pairwise Q-conj.)
$\Longrightarrow \lambda_j (\underline{d}_j^T Q \underline{d}_j) = 0 \Longrightarrow \underline{d}_j^T Q \underline{d}_j > 0$ (since Q is def.-pos. and $\underline{d}_j \neq \underline{0}$)
$\Longrightarrow \lambda_j = 0 \; \forall j$ ■

Geometric/algebraic interpretation

- If Q is diagonal (level curves are ellipses whose axes are aligned with the coordinate directions) $q(\underline{x})$ can be minimized via 1-D minimizations along those directions.

Illustration: Nocedal and Wright, Numerical Optimization, Edition 1999, p. 104.



if we start from $\forall \underline{x}_0$ we can get the optimal solution $\underline{x}^*$ by just optimizing along the directions parallel to the axes (Q is diagonal here)

- If Q is not diagonal, we exploit conjugate directions:

If $\underline{d}_0, \ldots, \underline{d}_{n-1}$ are n mutually Q-conjugate directions, linear variable transformation:

$$\underline{x} = \sum_{i=0}^{n-1} \alpha_i \underline{d}_i. \qquad : \text{we can write } \forall \underline{x} \text{ with } \{\underline{d}_k\}_k$$

The quadratic function becomes:

$$\Longrightarrow \tilde{q}(\underline{\alpha}) = \frac{1}{2}\Big(\sum_{i=0}^{n-1} \alpha_i \underline{d}_i\Big)^t Q \Big(\sum_{i=0}^{n-1} \alpha_i \underline{d}_i\Big) - \underline{b}^t \Big(\sum_{i=0}^{n-1} \alpha_i \underline{d}_i\Big) = \sum_{i=0}^{n-1} \Big[\frac{1}{2}\alpha_i^2 \underline{d}_i^t Q \underline{d}_i - \alpha_i \underline{b}^t \underline{d}_i\Big] = \sum_{i=0}^{n-1} \tilde{q}_i(\alpha_i).$$

where each $\tilde{q}_i$ is quadratic with single variable α_i.

Sum of 1 dimensional function, we can minimize them one by one

Minimization of $q(\underline{x})$ over $\mathbb{R}^n$ reduces to at most n 1-D minimization problems over $\mathbb{R}$.

Illustration: Nocedal and Wright, Numerical Optimization, Edition 1999, p. 105.

Theorem: (Conjugate directions)

Let $\{\underline{d}_j\}_{i=0}^{n-1}$ be n nonzero mutually Q-conjugate directions. For any $\underline{x}_0 \in \mathbb{R}^n$, $\{\underline{x}_k\}_{k \ge 0}$ generated according to

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \qquad (1)$$

with

$$\alpha_k = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k} \quad \text{and} \quad \underline{g}_k := \nabla q(\underline{x}_k) = Q\underline{x}_k - \underline{b}$$

terminates to the (unique) global optimal solution $\underline{x}^*$ of $q(\underline{x})$ in at most n iterations, that is

$$\underline{x}_n = \underline{x}_0 + \sum_{k=0}^{n-1} \alpha_k \underline{d}_k = \underline{x}^*.$$

Proof.:

Since $\underline{d}_k$'s are linearly independent, $\exists \alpha_k$'s such that

$$\underline{x}^* - \underline{x}_0 = \alpha_0 \underline{d}_0 + \ldots + \alpha_{n-1} \underline{d}_{n-1}.$$

Taking the scalar product with $\underline{d}_k^t Q$, we obtain

$$\alpha_k = \frac{\underline{d}_k^t Q(\underline{x}^* - \underline{x}_0)}{\underline{d}_k^t Q \underline{d}_k}.$$

Following the iterative process (1) from $\underline{x}_0$ to $\underline{x}_k$:

$$\underline{x}_k - \underline{x}_0 = \alpha_0 \underline{d}_0 + \ldots + \alpha_{k-1} \underline{d}_{k-1}$$

and hence Q-orthogonality of the $\underline{d}_k$'s implies that $\underline{d}_k^t Q(\underline{x}_k - \underline{x}_0) = 0$.

Therefore

$$\alpha_k = \frac{\underline{d}_k^t Q(\underline{x}^* - \underline{x}_k + \underline{x}_k - \underline{x}_0)}{\underline{d}_k^t Q \underline{d}_k} = \frac{\underline{d}_k^t Q(\underline{x}^* - \underline{x}_k)}{\underline{d}_k^t Q \underline{d}_k} = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}$$

since $\underline{g}_k = \nabla q(\underline{x}_k) = Q\underline{x}_k - \underline{b}$ and $Q\underline{x}^* = \underline{b}$. □

Property: (Expanding subspace)

Let $\underline{d}_0, \ldots, \underline{d}_{n-1}$ be nonzero mutually Q-conjugate vectors. Then, for any $\underline{x}_0 \in \mathbb{R}^n$, $\{\underline{x}_k\}_{k \geq 0}$ generated according to

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \quad \text{with} \quad \alpha_k = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}$$

is such that

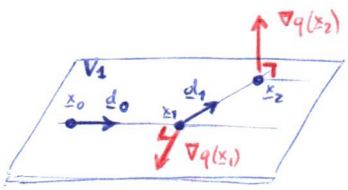
$$\underline{x}_k = \underline{x}_0 + \sum_{j=0}^{k-1} \alpha_j \underline{d}_j$$

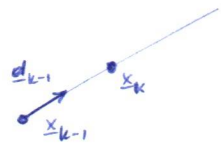
minimizes $q(\underline{x}) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{b}^t \underline{x}$ not only on the line

$$\{\, \underline{x} \in \mathbb{R}^n \,:\, \underline{x} = \underline{x}_{k-1} + \alpha \underline{d}_{k-1},\ \alpha \in \mathbb{R} \,\}$$

but also on the affine subspace $V_k = \{\, \underline{x} \in \mathbb{R}^n \,:\, \underline{x} = \underline{x}_0 + span\{\underline{d}_0, \ldots, \underline{d}_{k-1}\} \,\}$.

In particular, $\underline{x}_n$ is the global optimum of $q(\underline{x})$ on $\mathbb{R}^n$.





The theorem is telling us that $\underline{x}_2$ is optimal in the plane V_1 (span of $\underline{d}_0, \underline{d}_1$) since if we take the gradient of $q(\underline{x})$ in $\underline{x}_2$ this gradient goes out of the plane V_1 (perpendicularly)

⇓

At each iteration (k) we get an optimal solution for the subspace V_{k-1}

Proof: Since $q(\underline{x})$ is (strictly) convex, $\underline{x}^*$ is the global minimum of $q(\underline{x})$ on convex $C \subseteq \mathbb{R}^n$ if and only if $\nabla^t q(\underline{x}^*)(\underline{x} - \underline{x}^*) \geq \underline{0}$ for every $\underline{x} \in C$.

It suffices to verify that $\nabla q(\underline{x}_k) = Q\underline{x}_k - \underline{b}$ is orthogonal to V_k for each k, $1 \leq k \leq n$. (exercise 7.4) □

Consequence: In conjugate direction method the gradients $\underline{g}_k$ satisfy $\underline{g}_k^t \underline{d}_i = 0$ for all i with $1 \leq i < k$.

⇓

the gradient is always ⊥ to all the previous directions

How can we generate these n mutually Q-conjugate directions?
Conjugate GRADIENT method.

4.7.1 Conjugate gradient method for quadratic functions

Initialization: Arbitrary $\underline{x}_0$, $\underline{g}_0 = \nabla q(\underline{x}_0) = Q\underline{x}_0 - \underline{b}$, $\underline{d}_0 := -\underline{g}_0$ and $k = 0$

Iteration: $\underline{x}_{k+1} := \underline{x}_k + \alpha_k \underline{d}_k$ with $\alpha_k = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}$ (exact 1-D search)

$$\underline{d}_{k+1} := -\underline{g}_{k+1} + \beta_k \underline{d}_k \quad \text{with} \quad \beta_k = \frac{\underline{g}_{k+1}^t Q \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}.$$

basically it's a linear combination of the gradient and the previous direction

Observations:

- $\alpha_k = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}$ minimizes $q(\underline{x})$ along line through $\underline{x}_k$ generated by $\underline{d}_k$

$$\frac{dq(\underline{x}_k + \alpha \underline{d}_k)}{d\alpha} = \underline{d}_k^t Q(\underline{x}_k + \alpha \underline{d}_k) - \underline{b}^t \underline{d}_k = \underline{d}_k^t Q \underline{x}_k + \alpha \underline{d}_k^t Q \underline{d}_k - (-\nabla^t q(\underline{x}_k) + \underline{x}_k^t Q)\underline{d}_k = 0$$

- Limited computational requirements, no matrix inversions are needed.

To show that this algorithm finds the global optimal solution of $q(\underline{x})$ after at most n iterations, just verify that the generated directions are mutually Q-conjugated.

Proposition:

At each iteration k in which the optimum solution of $q(\underline{x})$ has not yet been found ($\underline{g}_i \neq \underline{0}$ for $i = 0, \ldots, k$)

i) $\underline{d}_0, \ldots, \underline{d}_{k+1}$ generated are mutually Q-conjugate

ii) $\alpha_k = \frac{\underline{g}_k^t \underline{g}_k}{\underline{d}_k^t Q \underline{d}_k} \neq 0$

iii) $\beta_k = \frac{\underline{g}_{k+1}^t (\underline{g}_{k+1} - \underline{g}_k)}{\underline{g}_k^t \underline{g}_k} = \frac{\underline{g}_{k+1}^t \underline{g}_{k+1}}{\underline{g}_k^t \underline{g}_k}$ ← we obtained an expression for β's that is ⊥ from Q. This is good for when we'll extend the method for arbitrary functions, where we don't necessarily have a constant positive definite Hessian matrix

Proof:

i) By induction on k, assuming that $\underline{d}_0, \ldots, \underline{d}_k$ are mutually Q-conjugate.

Clearly true for $k = 0$.

Show that $\underline{d}_{k+1}^t Q \underline{d}_i = 0 \ \ \forall i = 0, \ldots, k$.

For $i = k$ we have

$$\underline{d}_{k+1}^t Q \underline{d}_k = [-\underline{g}_{k+1}^t + \beta_k \underline{d}_k^t] Q \underline{d}_k = -\underline{g}_{k+1}^t Q \underline{d}_k + \beta_k \underline{d}_k^t Q \underline{d}_k$$

$$= -\underline{g}_{k+1}^t Q \underline{d}_k + \left(\frac{\underline{g}_{k+1}^t Q \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}\right) \underline{d}_k^t Q \underline{d}_k = -\underline{g}_{k+1}^t Q \underline{d}_k + \underline{g}_{k+1}^t Q \underline{d}_k = 0.$$

Verify that $\underline{d}_{k+1}^t Q \underline{d}_i = 0 \ \ \forall i = 0, \ldots, k-1$.

$\underline{d}_{k+1}^t Q \underline{d}_i = -\underline{g}_{k+1}^t Q \underline{d}_i + \beta_k \underline{d}_k^t Q \underline{d}_i$ with the induction assumption $\underline{d}_k^t Q \underline{d}_i = 0$ $\forall i = 0, \ldots, k-1$.

Since $\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{d}_i$ with $\alpha_i \neq 0$ (otherwise $\underline{x}_i$ is the optimal solution)

$$Q \underline{d}_i = \frac{1}{\alpha_i}(Q \underline{x}_{i+1} - Q \underline{x}_i) = \frac{1}{\alpha_i}(Q \underline{x}_{i+1} - \underline{b} - Q \underline{x}_i + \underline{b}) = \frac{1}{\alpha_i}(\underline{g}_{i+1} - \underline{g}_i)$$

$$= \frac{1}{\alpha_i}((-\underline{d}_{i+1} + \beta_i \underline{d}_i) - (-\underline{d}_i + \beta_{i-1} \underline{d}_{i-1}))$$

is a linear combination of $\underline{d}_{i+1}, \underline{d}_i$ and $\underline{d}_{i-1}$ ($Q \underline{d}_0$ only of $\underline{d}_1$ and $\underline{d}_0$, with $\underline{g}_0 = -\underline{d}_0$).

Now $\underline{d}_0, \ldots, \underline{d}_k$ mutually Q-conjugate implies (expanding subspace property) that $\underline{x}_{k+1}$ minimizes $q(\underline{x})$ on the subspace V_{k+1} generated by $\underline{d}_0, \ldots, \underline{d}_k$, with $\underline{x}_0 \in V_{k+1}$.

Therefore $\underline{g}_{k+1} = \nabla q(\underline{x}_{k+1})$ is orthogonal to V_{k+1} and $Q \underline{d}_i \in V_{k+1}$ for $i = 0, \ldots, k-1 \Rightarrow$ $\underline{g}_{k+1}^t Q \underline{d}_i = 0$.

ii) Since $\underline{d}_k = -\underline{g}_k + \beta_{k-1} \underline{d}_{k-1}$, $\alpha_k = -\frac{\underline{g}_k^t \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k}$ we can rewrite

$$\alpha_k = \frac{\underline{g}_k^t \underline{g}_k}{\underline{d}_k^t Q \underline{d}_k} - \beta_{k-1} \frac{\underline{g}_k^t \underline{d}_{k-1}}{\underline{d}_k^t Q \underline{d}_k}.$$

But $\underline{g}_k^t \underline{d}_{k-1} = 0$ since $\underline{d}_0, \ldots, \underline{d}_{k-1}$ Q-conjugate $\Rightarrow$ $\underline{x}_k$ minimum of $q(\underline{x})$ on V_k generated by $\underline{d}_0, \ldots, \underline{d}_{k-1}$ with $\underline{x}_0 \in V_k$; moreover $\alpha_k \neq 0$ since $\underline{g}_k \neq \underline{0}$.

iii) $\underline{g}_{k+1} - \underline{g}_k = Q(\underline{x}_{k+1} - \underline{x}_k) = \alpha_k Q \underline{d}_k$ implies that

$$\underline{g}_{k+1}^t Q \underline{d}_k = \frac{1}{\alpha_k} \underline{g}_{k+1}^t (\underline{g}_{k+1} - \underline{g}_k).$$

According to *ii*)

$$\beta_k = \frac{\underline{g}_{k+1}^t Q \underline{d}_k}{\underline{d}_k^t Q \underline{d}_k} = \frac{1}{\alpha_k} \frac{\underline{g}_{k+1}^t (\underline{g}_{k+1} - \underline{g}_k)}{\underline{d}_k^t Q \underline{d}_k} = \frac{\underline{d}_k^t Q \underline{d}_k}{\underline{g}_k^t \underline{g}_k} \frac{\underline{g}_{k+1}^t (\underline{g}_{k+1} - \underline{g}_k)}{\underline{d}_k^t Q \underline{d}_k} = \frac{\underline{g}_{k+1}^t (\underline{g}_{k+1} - \underline{g}_k)}{\underline{g}_k^t \underline{g}_k}.$$

Since $\underline{g}_k = -\underline{d}_k + \beta_{k-1} \underline{d}_{k-1}$ belongs to the subspace generated by $\underline{d}_o, \ldots, \underline{d}_k$ and $\underline{g}_{k+1}$ is orthogonal to this subspace, we have $\underline{g}_{k+1}^t \underline{g}_k = 0$ and hence

$$\beta_k = \frac{\underline{g}_{k+1}^t \underline{g}_{k+1}}{\underline{g}_k^t \underline{g}_k}. \qquad \square$$

Advantages: No need for matrix inversions, limited computational requirements.

Disadvantages:

- Exact or at least accurate 1-D search otherwise the directions may loose Q-conjugacy.
- The method is not invariant with respect to affine transformations of the coordinates.

In **fourth computer laboratory** compare the convergence speed of the gradient, conjugate gradient and Newton methods.

4.7.2 Conjugate direction methods (arbitrary functions)

For arbitrary functions with large n, α_k is determined in an approximate (inexact) way and β_k must not depend of Hessian.

Arbitrary $\underline{x}_0$ and $\underline{d}_0 = -\nabla f(\underline{x}_0)$

$$\underline{x}_{k+1} := \underline{x}_k + \alpha_k \underline{d}_k \quad \text{with inexact 1-D search and} \quad \underline{d}_{k+1} = -\nabla f(\underline{x}_{k+1}) + \beta_k \underline{d}_k.$$

Most popular formulae for β_k:

$$\beta_k^{FR} = \frac{\|\nabla f(\underline{x}_{k+1})\|^2}{\|\nabla f(\underline{x}_k)\|^2} \quad \text{Fletcher-Reeves} \longrightarrow \text{equivalent to: } \frac{\underline{g}_{k+1}^T \underline{g}_{k+1}}{\underline{g}_k^T \underline{g}_k} = \beta_k \text{ (proposition)}$$

preferred

$$\beta_k^{PR} = \frac{\nabla^t f(\underline{x}_{k+1})(\nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k))}{\|\nabla f(\underline{x}_k)\|^2} \quad \text{Polak-Ribière}$$

Observation: $\underline{d}_k$ is a descent direction if 1-D search is exact

$$\nabla^t f(\underline{x}_k)\underline{d}_k = -\|\nabla^t f(\underline{x}_k)\|^2 + \beta_{k-1}\nabla^t f(\underline{x}_k)\underline{d}_{k-1} = -\|\nabla^t f(\underline{x}_k)\|^2 < 0.$$

For quadratic functions the method coincides with CG method. (conjugate gradient)

For nonquadratic functions, Polak-Ribière version turns out to be more efficient than Fletcher-Reeves one.

Possibile explanation: nonquadratic terms and inexact 1-D search may progressively lead to loose Q-conjugacy.

Observations

- At each iteration it suffices to store $\underline{x}_k$, $\nabla f(\underline{x}_k)$, $\nabla f(\underline{x}_{k+1})$ and $\underline{d}_k$ (only 4 vectors of dimension n).
- Version with "restart" in which every m iterations ($m << n$) we set $\beta_k = 0$ (and hence $\underline{d}_{k+1} = -\nabla f(\underline{x}_{k+1})$) is globally convergent. !

  In practice, for large scale problems we hope to find a solution way before n iterations! With $\beta_k = 0$ all the previously acquired information is lost.

4.7.3 Convergence

Convergence for quadratic functions (quadratic strictly convex functions)

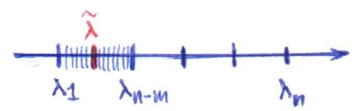
Let $q(\underline{x}) = \frac{1}{2}\underline{x}^t Q \underline{x} - \underline{b}^t \underline{x}$ be quadratic strictly convex with $\lambda_1 \leq \ldots \leq \lambda_n$ the eigenvalues of Q, then

$$\|\underline{x}_{k+1} - \underline{x}^*\|_Q^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|\underline{x}_0 - \underline{x}^*\|_Q^2$$

differences with the gradient method

where $\|\underline{x} - \underline{x}^*\|_Q^2 = (\underline{x}^t - \underline{x}^*)^t Q(\underline{x} - \underline{x}^*) = 2(q(\underline{x}) - q(\underline{x}^*))$.

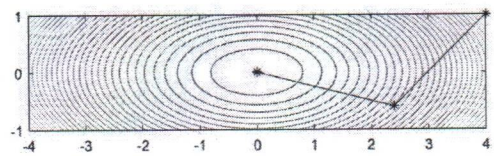
If m large eigenvalues and other $n - m$ "concentrated" around a $\tilde{\lambda}$, after $m + 1$ iterations $\|\underline{x}_{m+1} - \underline{x}^*\|_Q \approx \varepsilon \|\underline{x}_0 - \underline{x}^*\|_Q$ with $\varepsilon = (\lambda_{n-m} - \lambda_1)/2\tilde{\lambda}$, that is, we have an accurate estimate of the solution after $m + 1$ iterations.



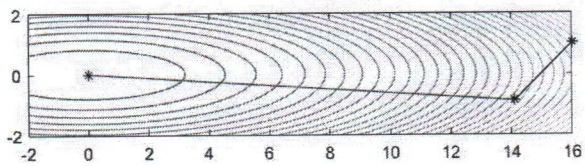
It means that if we have m large eigenvalues and the other ones are concentrated then after $m+1$ iterations we have a substancial improvement of the solution (this does not mean that we can stop after $m+1$ iterations)

Example:

$\min f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{a}{2}x_2^2$ with $a \geq 1$ and hence eigenvalues $\frac{1}{2}$ and $\frac{a}{2}$



$a = 4$



$a = 16$

The sequence $\{\underline{x}_k\}$ for $a = 4$ (top) and $a = 16$ (bottom), starting from $\underline{x}_0 = \begin{pmatrix} a \\ 1 \end{pmatrix}$.

Here we are in 2 dimensional space so we have the optimal solution in at most 2 iterations (!! on how elongated are the level curves (mathematically: how large is the spectrum between eigenvalues))

Convergence for arbitrary functions

1) If $f \in C^2$ and $\{\underline{x}_k\}_{k\geq 0}$ generated by the F-R method with exact 1-D search converges to $\underline{x}^*$ with p.d. $H(\underline{x}^*)$, then

$$\lim_{k\to\infty} \frac{\|\underline{x}_{k+n} - \underline{x}^*\|}{\|\underline{x}_k - \underline{x}^*\|} = 0,$$

namely convergence is superlinear within n iterations.

→ we have superlinear convergence in n iterations (only "superlinear convergence" would be $n = 1$)

Similar result also for inexact 1-D search.

2) Global convergence of F-R method even without "restart" (not known for P-R).

Zoutendijk's theorem implies: For F-R method with inexact 1-D search satisfying the strong Wolfe conditions with $0 < c_1 < c_2 < 1/2$, we have

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

A sub-sequence has the gradient norm that converges to 0 (the smallest limit point is 0).

4.7.4 Preconditioned conjugate gradient method

The conjugate gradient (CG) method can be accelerated by a variable change $\underline{x} = S\underline{y}$, where S is $n \times n$ symmetric and non singular.

→ we know that the speed of convergence depends on the spectrum, if we do an appropriate variable change then we can accelerate the method (accelerate the speed of convergence)

S: matrix of the change of coordinates

By applying CG method to the equivalent function

$$h(\underline{y}) = q(S\underline{y}) = \frac{1}{2}\underline{y}^t SQS\underline{y} - \underline{b}^t S\underline{y}$$

we obtain

$$\underline{y}_{k+1} = \underline{y}_k + \alpha_k \tilde{\underline{d}}_k$$

with α_k determined by 1-D search, $\tilde{\underline{d}}_0 = -\nabla h(\underline{y}_0)$ and $\tilde{\underline{d}}_k = -\nabla h(\underline{y}_k) + \beta_{k-1}\tilde{\underline{d}}_{k-1}$ for $k = 1, \ldots, n-1$ where

$$\beta_{k-1} = \frac{\nabla^t h(\underline{y}_k)\nabla h(\underline{y}_k)}{\nabla^t h(\underline{y}_{k-1})\nabla h(\underline{y}_{k-1})}. \quad (F\text{-}R)$$

Setting $\underline{x}_k = S\underline{y}_k$, $\nabla h(\underline{y}_k) = S\underline{g}_k$, $\underline{d}_k = S\tilde{\underline{d}}_k$, we obtain the equivalent **preconditioned conjugate gradient method**:

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k$$

with α_k determined by 1-D search, $\underline{d}_0 = -S^2\underline{g}_0$ and

$$\underline{d}_k = -S^2\underline{g}_k + \beta_{k-1}\underline{d}_{k-1} \quad \text{for } k = 1, \ldots, n-1$$

where

$$\beta_{k-1} = \frac{\underline{g}_k^t S^2 \underline{g}_k}{\underline{g}_{k-1} S^2 \underline{g}_{k-1}}.$$

Clearly when $S = I$ it coincides with the standard CG method.

Since $\nabla^2 h(\underline{y}) = SQS$, the directions $\tilde{\underline{d}}_0, \ldots \tilde{\underline{d}}_{n-1}$ are (SQS)-conjugate. Moreover $\underline{d}_k = S\tilde{\underline{d}}_k$ implies that $\underline{d}_0, \ldots \underline{d}_{n-1}$ are Q-conjugate.

As the convergence speed depends on eigenvalues of SQS, we look for S such that SQS has a smaller condition number than Q or eigenvalues that are distributed into "groups".

Recall: a good approximate solution can be found in a number of iterations not much larger than the number of groups.

Gradient method: globally convergent, light computationally but extremely slow

Newton method: locally convergent, very fast (quadratic when we're sufficiently close) but heavy computationally

large number of variables → **conjugate directions method** (ex. conj. gradient method): much lighter than Newton and they're super-linear within n steps

We try to derive variants of the Newton method where we do not directly deal with the second order derivative information but we extract this information (/approximation) from variation in the gradient of the function and so we avoid to invert the Hessian matrix (in fact we don't use it)

4.8 Quasi-Newton methods

Variants of Newton method where 2nd order derivative information is extracted from variations in $\nabla f(\underline{x})$ rather than by using/inverting $\nabla^2 f(\underline{x}_k)$.

Generate $\{H_k\}$ of symmetric p.d. approximations of $[\nabla^2 f(\underline{x}_k)]^{-1}$ and take

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \quad \text{with} \quad \underline{d}_k = -H_k \nabla f(\underline{x}_k).$$

where $\alpha_k > 0$ minimizes $f(\underline{x})$ along $\underline{d}_k$ or satisfies inexact search conditions.

Advantages w.r.t. Newton method:

- since H_k's are symmetric and p.d., always well defined and descent direction,
- only involves first order derivatives,
- H_k is constructed iteratively, each iteration is $O(n^2)$.

Disadvantages w.r.t. conjugate direction methods: requires storing/handling matrices.

Idea: Second order derivative information is extracted from $\nabla f(\underline{x}_k)$ and $\nabla f(\underline{x}_{k+1})$.

Quadratic approximation of $f(\underline{x})$ around $\underline{x}_k$:

$$f(\underline{x}_k + \underline{\delta}) \approx f(\underline{x}_k) + \underline{\delta}^t \nabla f(\underline{x}_k) + \frac{1}{2}\underline{\delta}^t \nabla^2 f(\underline{x}_k)\underline{\delta}. \qquad \underline{\delta} = \text{step}$$

Differentiating we obtain (w.r.t. $\underline{\delta}$):

$$\nabla f(\underline{x}_k + \underline{\delta}) \approx \nabla f(\underline{x}_k) + \nabla^2 f(\underline{x}_k)\underline{\delta}.$$

Substituting $\underline{\delta}$ with $\underline{\delta}_k$ and setting $\underline{\delta}_k = \underline{x}_{k+1} - \underline{x}_k$ and $\underline{\gamma}_k = \nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k)$ we have

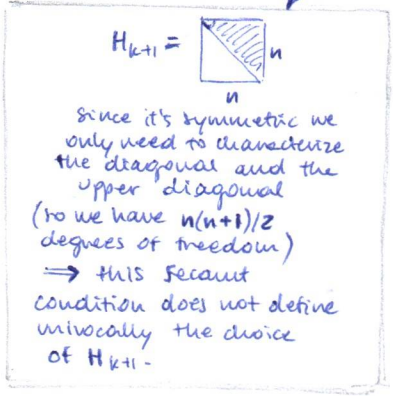
$$\underline{\gamma}_k \approx \nabla^2 f(\underline{x}_k)\underline{\delta}_k, \quad \text{namely} \quad [\nabla^2 f(\underline{x}_k)]^{-1}\underline{\gamma}_k \approx \underline{\delta}_k.$$

we have a relationship between the increment of $\nabla f(\cdot)$ and the increment of $\underline{x}$ (the Hessian matrix links them)

Since $\underline{\delta}_k$ and $\underline{\gamma}_k$ can only be determined after 1-D search, we select H_{k+1} symmetric and p.d. such that

$$H_{k+1}\underline{\gamma}_k = \underline{\delta}_k \quad \textbf{(secant condition)}. \tag{1}$$

H_{k+1} is not univocally defined: n equations and $n(n+1)/2$ degrees of freedom.



Since it's symmetric we only need to characterize the diagonal and the upper diagonal (so we have $n(n+1)/2$ degrees of freedom) ⟹ this secant condition does not define univocally the choice of H_{k+1}.

How we can construct a sequence $\{H_k\}$ by looking for solution of the secant condition? A good way is to proceede by successive updates: $H_0 \to H_1 \to H_2 \to \dots$ we don't want a method that is heavy computationally (ex. $H_0 \to H_1$, $(H_0, H_1) \to H_2$, $(H_0, H_1, H_2) \to H_3$, ...)

Simple way is by *successive updates*:

$$H_{k+1} = H_k + a_k \underline{u}\,\underline{u}^t \tag{2}$$

where $\underline{u}\,\underline{u}^t$ symmetric matrix of rank 1 and a_k proportionality coefficient.
(the rows are linearly dependent)

To satisfy (1) we must have

$$H_k\underline{\gamma}_k + a_k \underline{u}\,\underline{u}^t\underline{\gamma}_k = \underline{\delta}_k$$

and hence $\underline{u}$ and $(\underline{\delta}_k - H_k\underline{\gamma}_k)$ must be collinear.

Since a_k accounts for proportionality, we can set $\underline{u} = \underline{\delta}_k - H_k\underline{\gamma}_k$ and hence $a_k\underline{u}^t\underline{\gamma}_k = 1$.

Rank one update formula:

$$H_{k+1} = H_k + \frac{(\underline{\delta}_k - H_k\underline{\gamma}_k)(\underline{\delta}_k - H_k\underline{\gamma}_k)^t}{(\underline{\delta}_k - H_k\underline{\gamma}_k)^t\underline{\gamma}_k} \tag{3}$$

it allows us to generate a sequence of matrices starting from a matrix H_0 by just adding at each iteration a rank 1 matrix to it. This sequence satisfies the secant condition at each iteration

Properties

1. For quadratic strictly convex functions, $H_n = Q^{-1}$ in at most n iterations, even with inexact 1-D search. (Q^{-1}: inverse of the Hessian matrix)
2. !! No guarantee that H_k is p.d.!

The rank one update is not enough in a reuse, so:

Rank two updates

$$H_{k+1} = H_k + a_k \underline{u}\,\underline{u}^t + b_k \underline{v}\,\underline{v}^t \tag{4}$$

are more interesting.

we add to matrices of rank 1

To satisfy (1) we have

$$H_k\underline{\gamma}_k + a_k \underline{u}\,\underline{u}^t\underline{\gamma}_k + b_k \underline{v}\,\underline{v}^t\underline{\gamma}_k = \underline{\delta}_k$$

where $\underline{u}, \underline{v}$ are not determined univocally.

Setting $\underline{u} = \underline{\delta}_k$ and $\underline{v} = H_k\underline{\gamma}_k$, we obtain $a_k\underline{u}^t\gamma_k = 1$ and $b_k\underline{v}^t\gamma_k = -1$
and hence the rank two update formula:

$$H_{k+1} = H_k + \frac{\underline{\delta}_k\underline{\delta}_k^t}{\underline{\delta}_k^t\underline{\gamma}_k} - \frac{H_k\underline{\gamma}_k\underline{\gamma}_k^t H_k}{\underline{\gamma}_k^t H_k\underline{\gamma}_k} \tag{5}$$

Davidon-Fletcher-Powell (DFP)

Proposition: If $\underline{\delta}_k^t\underline{\gamma}_k > 0 \quad \forall k$ (curvature condition),
the DFP method preserves the positive definiteness of H_k, i.e., if H_0 is p.d. then H_k is p.d. for all $k \geq 1$.

Proof: Suppose H_0 is p.d. and proceed by induction.

Verify that if H_k is p.d. then $\underline{z}^t H_{k+1}\underline{z} > 0 \; \forall \underline{z} \neq \underline{0}$. (i.e. H_k pos. definite ⟹ H_{k+1} positive definite)

If H_k is p.d. it admits a Cholesky factorization $H_k = L_k L_k^t$. → $H_k = L_k L_k^T$: $L_k =$ ◣

Eliminating subscripts k and setting $\underline{a} = L^t\underline{z}$ and $\underline{b} = L^t\underline{\gamma}$ we have

$$\underline{z}^t(H - \frac{H\underline{\gamma}\,\underline{\gamma}^t H}{\underline{\gamma}^t H\underline{\gamma}})\underline{z} = \underline{a}^t\underline{a} - \frac{(\underline{a}^t\underline{b})^2}{\underline{b}^t\underline{b}} \geq 0$$

because $|\underline{a}^t\underline{b}| \leq \|\underline{a}\|\|\underline{b}\|$ (Cauchy-Schwarz inequality).

Since $\underline{z} \neq \underline{0}$, equality only holds if $\underline{a}$ and $\underline{b}$ collinear, namely if $\underline{z}$ and $\underline{\gamma}$ are collinear.

Since $\underline{\delta}^t\underline{\gamma} > 0$ we have that

$$\underline{z}^t(\frac{\underline{\delta}\,\underline{\delta}^t}{\underline{\delta}^t\underline{\gamma}})\underline{z} \geq 0$$

which holds in strict sense if $\underline{z}$ and $\underline{\gamma}$ are collinear.

Then just "develop" $\underline{z}^t H_{k+1}\underline{z}$ and apply the two inequalities. □

Fact: The curvature condition $\underline{\delta}_k^t \underline{\gamma}_k > 0$ holds for every $k \geq 0$ provided that the 1-D search satisfies (weak or strong) Wolfe conditions.

= the curvature condition is automatically satisfied if we solve the inexact 1D search with Wolfe condition

Proof:

For quadratic strictly convex functions, $\underline{\gamma}_k = Q\underline{\delta}_k$ implies $\underline{\delta}_k^t Q \underline{\delta}_k = \underline{\delta}_k^t \underline{\gamma}_k > 0$ because Q is p.d.

For arbitrary functions:

Weak Wolfe conditions

$$f(\underline{x}_k + \alpha_k \underline{d}_k) \leq f(\underline{x}_k) + c_1 \alpha_k \nabla^t f(\underline{x}_k)\underline{d}_k \quad \text{(Armijo criterion)} \tag{6}$$

$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0)$

$$\nabla^t f(\underline{x}_k + \alpha_k \underline{d}_k)\underline{d}_k \geq c_2 \nabla^t f(\underline{x}_k)\underline{d}_k \tag{7}$$

$\phi'(\alpha) \geq c_2 \phi'(0)$

with $0 < c_1 < c_2 < 1$.

Since $\underline{\delta}_k = \alpha_k \underline{d}_k$, (7) implies that

($\underline{d}_k \neq \underline{0}$)

$$\nabla^t f(\underline{x}_{k+1})\underline{\delta}_k \geq c_2 \nabla^t f(\underline{x}_k)\underline{\delta}_k$$

which in turn implies that

$$\underline{\gamma}_k^t \underline{\delta}_k \geq (c_2 - 1)\alpha_k \nabla^t f(\underline{x}_k)\underline{d}_k$$

< 0 > 0 < 0

with $(c_2 - 1) < 0$, $\alpha_k > 0$, and $\nabla^t f(\underline{x}_k)\underline{d}_k < 0$ because $\underline{d}_k$ is a descent direction. □

$\Rightarrow \underline{\gamma}_k^T \underline{\delta}_k > 0$

Properties

For quadratic strictly convex functions, DFP method with exact 1-D search:

1. terminates in at most n iterations with $H_n = Q^{-1}$,
2. generates Q–conjugate directions (from $H_0 = I$ it generates CG directions),
3. secant condition is hereditary, i.e., $H_i \underline{\gamma}_j = \underline{\delta}_j$ for $j = 0, \ldots, i-1$.

For arbitrary functions:

4. if $\underline{\delta}_k^t \underline{\gamma}_k > 0$ (curvature condition), all H_k are p.d. if H_0 is p.d. (hence descent method),
5. each iteration is $O(n^2)$,
6. superlinear convergence rate (in general only local),
7. if $f(\underline{x})$ convex, DFP method with exact 1-D search is globally convergent.

BFGS method

better performance than Quasi-Newton methods

We can construct an approximation of $\nabla^2 f(\underline{x}_k)$ instead of $[\nabla^2 f(\underline{x}_k)]^{-1}$.

Since we aim at $B_k \approx \nabla^2 f(\underline{x}_k)$, B_k must satisfy $B_{k+1}\underline{\delta}_k = \underline{\gamma}_k$. ← secant condition adapted

Taking $B_{k+1} = B_k + a_k \underline{u}\,\underline{u}^t + b_k \underline{v}\,\underline{v}^t$, with similar manipulations, we have:

rank two update

$$B_{k+1} = B_k + \frac{\underline{\gamma}_k \underline{\gamma}_k^t}{\underline{\gamma}_k^t \underline{\delta}_k} - \frac{B_k \underline{\delta}_k \underline{\delta}_k^t B_k}{\underline{\delta}_k^t B_k \underline{\delta}_k} \tag{8}$$

Notice that: we can invert all the matrices B_k in closed form (analitically, not numerically, so in this way the method is not so computionally heavy)

which should be inverted at each iteration to obtain H_{k+1}.

By applying twice *Sherman–Morrison indentity*

$$(A + \underline{a}\,\underline{b}^t)^{-1} = A^{-1} - \frac{A^{-1}\underline{a}\,\underline{b}^t A^{-1}}{1 + \underline{b}^t A^{-1}\underline{a}}, \quad A \in \mathbb{R}^{n \times n} \text{ non singular, } \underline{a}, \underline{b} \in \mathbb{R}^n, \text{ denominator} \neq 0,$$

we obtain the Broyden Fletcher Goldfarb and Shanno (BFGS) update formula:

$$H_{k+1} = H_k + \left(1 + \frac{\underline{\gamma}_k^t H_k \underline{\gamma}_k}{\underline{\delta}_k^t \underline{\gamma}_k}\right) \frac{\underline{\delta}_k \underline{\delta}_k^t}{\underline{\delta}_k^t \underline{\gamma}_k} - \frac{H_k \underline{\gamma}_k \underline{\delta}_k^t + \underline{\delta}_k \underline{\gamma}_k^t H_k}{\underline{\delta}_k^t \underline{\gamma}_k} \tag{9}$$

Easy to verify that $B_{k+1} H_{k+1} = I$ if $B_k H_k = I$.

The BFGS method has same properties 1 to 5 as DFP method.

In pratice, it is more robust w.r.t. to rounding errors and inexact 1-D search.

BFGS and DFP are two extreme cases of unique Broyden family of update formulae:

$$H_{k+1} = (1 - \phi) H_{k+1}^{\text{DFP}} + \phi H_{k+1}^{\text{BFGS}}$$

with $0 \leq \phi \leq 1$.

Properties: (Broyden family)

- H_{k+1} satisfies secant condition and is p.d. if $\underline{\delta}_k^t \underline{\gamma}_k > 0$.
- Methods invariant w.r.t. affine variable transformations.
- If $f(\underline{x})$ quadratic strictly convex, methods with exact 1-D search find $\underline{x}^*$ in at most n iterations ($H_n = Q^{-1}$) and the generated directions are Q-conjugate.
- Quasi-Newton methods are much less "sensitive" to inexact 1-D search than CD methods.

conjugate directions

Convergence of quasi-Newton methods

Complex analysis because approximation of Hessian (inverse) is updated at each iteration.

Convergence speed for $\{B_k\}$ or $\{H_k\}$ with inexact 1-D search (Wolfe cond.) where $\alpha_k = 1$ is tried first:

it tells us how fast these sequences of matrices should converge to the inverse of the Hessian or to the Hessian in order to guarantee super linear convergence (it's a necessary and sufficient condition)

Theorem: (Dennis and Moré)

Consider $f \in \mathcal{C}^3$ and quasi-Newton method with B_k p.d. and $\alpha_k = 1$ for each k. If $\lim_{k\to\infty} \underline{x}_k = \underline{x}^*$ with $\nabla f(\underline{x}^*) = \underline{0}$ and $\nabla^2 f(\underline{x}^*)$ is p.d., $\{\underline{x}_k\}$ converges superlinearly if and only if

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(\underline{x}^*))\underline{d}_k\|}{\|\underline{d}_k\|} = 0. \qquad (10)$$

approximation of the Hessian matrix (not the inverse)

Superlinearity means that the rate tends to zero when $k \to \infty$

If quasi-Newton $\underline{d}_k$ approximates Newton direction well enough, $\alpha_k = 1$ satisfies Wolfe cond. when $\underline{x}_k \to \underline{x}^*$.

Observation: No need that $B_k \to \nabla^2 f(\underline{x}^*)$, it suffices that B_k's become increasingly accurate approximations of $\nabla^2 f(\underline{x}^*)$ along the directions $\underline{d}_k$!

The necessary and sufficient condition (10) is satisfied by quasi-Newton methods such as BFGS and DFP.

- Comparing the convergence rates of gradient, Newton and BFGS methods: (speed)

  example for Rosenbrock's function, see page 199 (Chapter 8) of J. Nocedal, S. Wright, Numerical Optimization, Springer, first edition, 1999.

- Global convergence:

  Under some assumptions, $\exists$ global convergence results for arbitrary functions with inexact 1-D search.

  In general "classical" globalization techniques (restart or trust region) are not adopted for quasi-Newton methods because no examples of non convergence are known.

  Widely used: quasi-Newton methods with BFGS and DFP updates and 1-D search procedures satisfying Wolfe conditions.

A naive implementation of this variant is not efficient for unconstrained minimization, because it requires the system $B_k p_k = -\nabla f_k$ to be solved for the step p_k, thereby increasing the cost of the step computation to $O(n^3)$. We discuss later, however, that less expensive implementations of this variant are possible by updating Cholesky factors of B_k.

PROPERTIES OF THE BFGS METHOD

It is usually easy to observe the superlinear rate of convergence of the BFGS method on practical problems. Below, we report the last few iterations of the steepest descent, BFGS, and an inexact Newton method on Rosenbrock's function (2.23). The table gives the value of $\|x_k - x^*\|$. The Wolfe conditions were imposed on the step length in all three methods. From the starting point $(-1.2, 1)$, the steepest descent method required 5264 iterations, whereas BFGS and Newton took only 34 and 21 iterations, respectively to reduce the gradient norm to 10^{-5}.

= gradient method

accuracy we're requiring

steep. desc.	BFGS	Newton
1.827e-04	1.70e-03	3.48e-02
1.826e-04	1.17e-03	1.44e-02
1.824e-04	1.34e-04	1.82e-04
1.823e-04	1.01e-06	1.17e-08

iterations it takes to achieve that accuracy

A few points in the derivation of the BFGS and DFP methods merit further discussion. Note that the minimization problem (8.15) that gives rise to the BFGS update formula does not explicitly require the updated Hessian approximation to be positive definite. It is easy to show, however, that H_{k+1} will be positive definite whenever H_k is positive definite, by using the following argument. First, note from (8.8) that $y_k^T s_k$ is positive, so that the updating formula (8.16), (8.17) is well-defined. For any nonzero vector z, we have

$$z^T H_{k+1} z = w^T H_k w + \rho_k (z^T s_k)^2 \geq 0,$$

where we have defined $w = z - \rho_k y_k (s_k^T z)$. The right hand side can be zero only if $s_k^T z = 0$, but in this case $w = z \neq 0$, which implies that the first term is greater than zero. Therefore, H_{k+1} is positive definite.

In order to obtain quasi-Newton updating formulae that are invariant to changes, in the variables, it is necessary that the objectives (8.9a) and (8.15a) be also invariant. The choice of the weighting matrices W used to define the norms in (8.9a) and (8.15a) ensures that this condition holds. Many other choices of the weighting matrix W are possible, each one of them giving a different update formula. However, despite intensive searches, no formula has been found that is significantly more effective than BFGS.

The BFGS method has many interesting properties when applied to quadratic functions. We will discuss these properties later on, in the more general context of the Broyden family of updating formulae, of which BFGS is a special case.

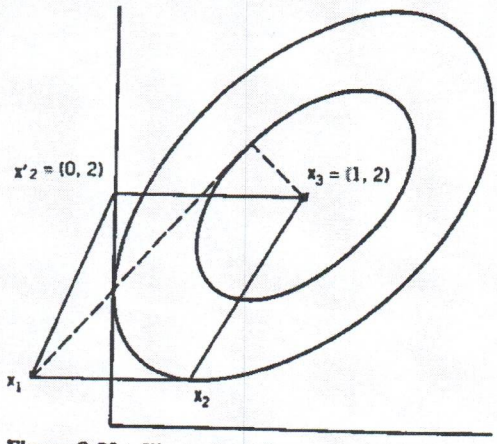


Figure 8.20 Illustration of conjugate directions.

Note that the Hessian matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$$

We now generate two conjugate directions, $\mathbf{d}_1$ and $\mathbf{d}_2$. Suppose we choose $\mathbf{d}_1' = (1, 0)$. Then, $\mathbf{d}_2' = (a, b)$ must satisfy $0 = \mathbf{d}_1'\mathbf{H}\mathbf{d}_2 = 8a - 4b$. In particular, we may choose $a = 1$ and $b = 2$ so that $\mathbf{d}_2' = (1, 2)$. It may be noted that the conjugate directions are not unique.

If we minimize the objective function f starting from $\mathbf{x}_1' = (-\frac{1}{2}, 1)$ along the direction $\mathbf{d}_1$, we get the point $\mathbf{x}_2' = (\frac{1}{2}, 1)$. Now, starting from $\mathbf{x}_2$ and minimizing along $\mathbf{d}_2$, we get $\mathbf{x}_3' = (1, 2)$. Note that $\mathbf{x}_3$ is the minimizing point.

The contours of the objective function and the path taken to reach the optimal point are shown in Figure 8.20. The reader can easily verify that, starting from any point and minimizing along $\mathbf{d}_1$ and $\mathbf{d}_2$, the optimal point is reached in, at most, two steps as shown, for example, by dashed lines in Figure 8.20. Furthermore, if we had started at $\mathbf{x}_1$ and then minimized along $\mathbf{d}_2$ first and next along $\mathbf{d}_1$, the optimizing step lengths along these respective directions would have remained the same as for the first case, taking the iterates from $\mathbf{x}_1$ to $\mathbf{x}_2' = (0, 2)'$ to $\mathbf{x}_3$.

Optimization of Quadratic Functions: Finite Convergence

The above example demonstrates that a quadratic function can be minimized in, at most, n steps, provided that we search along conjugate directions of the Hessian matrix. This result is generally true for quadratic functions, as shown by Theorem 8.8.3 below. This, coupled with the fact that a general function can be closely represented by its quadratic approximation in the vicinity of the optimal point, makes the notion of conjugacy very useful for optimizing both quadratic and nonquadratic functions. Note also that this result shows that if we start at $\mathbf{x}_1$, then, at each step $k = 1, \ldots, n$, the point $\mathbf{x}_{k+1}$ obtained minimizes f over the linear subspace containing $\mathbf{x}_1$ that is spanned by the vectors $\mathbf{d}_1, \ldots, \mathbf{d}_k$. Moreover, the gradient $\nabla f(\mathbf{x}_{k+1})$, if nonzero, is orthogonal to this subspace. This is sometimes called the *expanding subspace property*, and is illustrated in Figure 8.21 for $k = 1, 2$.