3.5 Conjugate directions method

- No derivative $\Rightarrow$ CC
- ∇f known $\Rightarrow$ SD

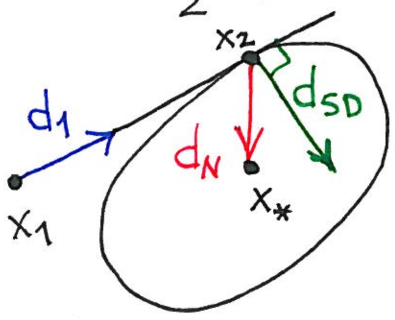
} zigzagging

- $\nabla f, H$ known $\Rightarrow$ N/MN - need H

We would like to find directions as "good" as in N, but without H.

Ex. $q(x) = \frac{1}{2} x^T H x + c^T x + d = [\text{Ex. } 1.56] =$

$$= \frac{1}{2}(x - x_*)^T H (x - x_*) + q_0$$



$d_{SD} = -\nabla q(x_2) \perp d_1$

$d_N = x_* - x$ "good"

$$\nabla q(x_2) = H(x_2 - x_*) = -H d_N \perp d_1 \Rightarrow$$

$$\Rightarrow \underline{\underline{d_1^T H d_N = 0}}.$$

A "good" direction should satisfy this condition!

Def. Vectors $d_1, d_2, \ldots, d_n \in \mathbb{R}^n$ are H-conjugate directions if

1) they are linearly independent,
2) $d_i^T H d_j = 0, \quad \forall\, i \neq j$.

Remark: define $n \times n$ matrix S

$$S = \begin{bmatrix} d_1 & \cdots & d_n \end{bmatrix}. \quad \text{Then}$$

$$S^T H S = \begin{bmatrix} d_1^T \\ \vdots \\ d_n^T \end{bmatrix} H \begin{bmatrix} d_1 & \cdots & d_n \end{bmatrix} =$$

$$= \begin{bmatrix} d_1^T H d_1 & d_1^T H d_2 & \cdots \\ d_2^T H d_1 & d_2^T H d_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} d_1^T H d_1 & & 0 \\ & d_2^T H d_2 & \\ 0 & & \ddots \end{bmatrix} =$$

$$= \Gamma \text{ - diagonal.}$$

Thus, $d_1, \ldots, d_n$ - H-conjugate $\Leftrightarrow$

$\Leftrightarrow$ $\det S \neq 0$, $S^T H S = \Gamma$ - diag.

<u>Remark</u>: if H pos. def. then

All $d_k \neq 0$ (nonzero vectors) $\Rightarrow$

$\Rightarrow d_k^T H d_k \neq 0 \Rightarrow \Gamma$ invertible $\Rightarrow$

$\Rightarrow S$ invertible, i.e. $d_1, \ldots, d_n$ - lin. indep.

<u>Special case for H pos. def.</u>

$d_1, \ldots, d_n$ - H-conj. $\Leftrightarrow$ $\begin{array}{l} S^T H S = \Gamma - \text{diag.} \\ d_1, \ldots, d_n \neq 0. \end{array}$

<u>Ex</u> Eigenvectors of H are H-conj. (Ex. 3.13, p. 89), but they are "hard" to find $\Rightarrow$ useless...

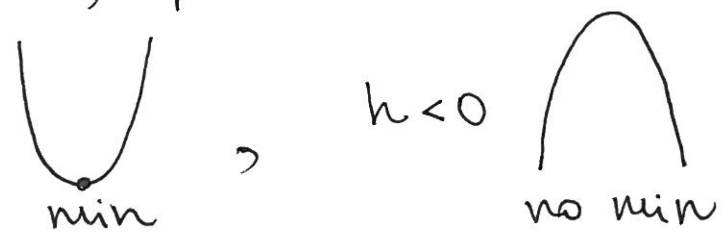
3.5.2. Minimization of a quadratic function.

$$q(x) = \frac{1}{2} x^T H x + c^T x + d$$

Assume that H - pos. def. ($\Rightarrow \exists \min$).

<u>Ex</u> $x \in \mathbb{R}$, $q(x) = h x^2$

$h > 0$: , $h < 0$



Rewrite as $q(x) = \frac{1}{2}(x - \bar{x})^T H (x - \bar{x}) + \bar{q}$.

<u>Here</u>: $\bar{x}$ is the minimum, $\bar{q} = \text{const}$.

(not important, can assume $\bar{q} = 0$).

Let $d_1, \ldots, d_n$ be H-conjugate $\Rightarrow$

$\Rightarrow$ it is a basis in $\mathbb{R}^n$.

$$x = \sum_{k=1}^{n} \alpha_k d_k = \underbrace{\begin{bmatrix} d_1 & \dots & d_n \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}}_{\alpha \text{ : new coordinates}} = S \cdot \alpha$$

$\bar{x} = S \cdot \bar{\alpha}$: the optimal point.

$$x - \bar{x} = S\alpha - S\bar{\alpha} = S(\alpha - \bar{\alpha}) \implies$$

$$\implies q(x) = \frac{1}{2}(x-\bar{x})^T H (x - \bar{x}) =$$

$$= \frac{1}{2}(\alpha - \bar{\alpha})^T \underbrace{S^T H S}_{\text{"}\Gamma = \begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_n \end{bmatrix}} (\alpha - \bar{\alpha}) =$$

$$= \frac{1}{2} \sum_{k=1}^{n} \overset{\overset{0}{\wedge}}{\gamma_i} (\alpha_i - \bar{\alpha}_i)^2 .$$

Let us apply the CC search with the basis $d_1, \dots, d_n$.

Along d_1: $\min_{\lambda} q(x_1 + \lambda d_1)$

$$x_1 + \lambda d_1 = S \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} + S \begin{bmatrix} \lambda \\ 0 \\ \vdots \\ 0 \end{bmatrix} = S \begin{bmatrix} \alpha_1 + \lambda \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} .$$

$$q(x_1 + \lambda d_1) = \frac{1}{2}\gamma_1 (\alpha_1 + \lambda - \bar{\alpha}_1)^2 +$$

$$+ \frac{1}{2}\gamma_2 (\alpha_2 - \bar{\alpha}_2)^2 + \dots + \frac{1}{2}\gamma_n (\alpha_n - \bar{\alpha}_n)^2 .$$

$$\min_{\lambda} : \boxed{\alpha_1 + \lambda_{min} - \bar{\alpha}_1 = 0 .}$$

The first term vanishes!

And we arrive at x_2

$$x_2 = x_1 + \lambda_{min} d_1 = S \begin{bmatrix} \bar{\alpha}_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Along d_2: $\min_{\lambda} q(x_2+\lambda d_2)$

$$x_2+\lambda d_2 = S\begin{bmatrix}\bar\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{bmatrix} + S\begin{bmatrix}0\\ \lambda\\ 0\\ \vdots\\ 0\end{bmatrix} = S\begin{bmatrix}\bar\alpha_1\\ \alpha_2+\lambda\\ \vdots\\ \alpha_n\end{bmatrix}$$

$$q(x_2+\lambda d_2) = 0 + \frac{1}{2}\gamma_2(\alpha_2+\lambda-\bar\alpha_2)^2 + $$
$$+\frac{1}{2}\gamma_3(\alpha_3-\bar\alpha_3)^2+\dots+\frac{1}{2}\gamma_n(\alpha_n-\bar\alpha_n)^2.$$

$\min_{\lambda}$: $\boxed{\alpha_2+\lambda_{min}-\bar\alpha_2=0} \Rightarrow$

$$\Rightarrow x_3 = x_2+\lambda_{min} d_2 = S\begin{bmatrix}\bar\alpha_1\\ \bar\alpha_2\\ \alpha_3\\ \vdots\\ \alpha_n\end{bmatrix}$$

After minimizing along all $d_1,\dots,d_n$:

$$x_{n+1} = S\begin{bmatrix}\bar\alpha_1\\ \bar\alpha_2\\ \vdots\\ \bar\alpha_n\end{bmatrix} = S\bar\alpha = \bar x$$

the optimal point.

Th (Theorem 4, p. 73)

$q(x) = \frac{1}{2}x^T H x + c^T x + d$, H pos. def.,

$d_1,\dots,d_n$ – H-conj. and $x_1\in\mathbb{R}^n$.

Then the algorithm $x_{k+1}=x_k+\lambda_k d_k$

where λ_k solves $\min_{\lambda} q(x_k+\lambda d_k)$

gives $x_{n+1}=\bar x$ – the minimum

point for $\min_{x\in\mathbb{R}^n} q(x)$.

Remark: for a quadratic function

"λ_k that solves $\min_{\lambda} q(x_k+\lambda d_k)$"

can be found explicitly (Ex. 3.1)

$$\lambda_k = \frac{d_k^T(Hx_k+c)}{d_k^T H d_k}.$$

(not working for a general function!)

How to find H-conjugate directions

a) By solving linear systems, e.g.

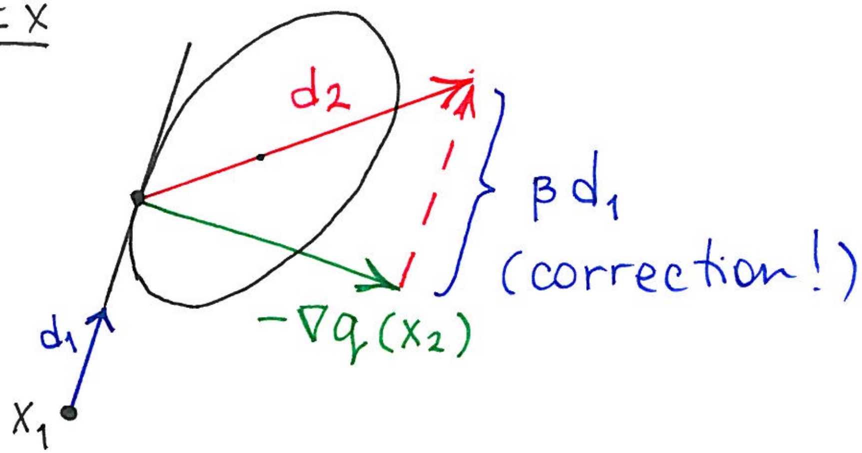
- $d_1 = -\nabla q(x_1)$,
- $d_1^T H d_2 = 0$, $\nabla q(x_2)^T d_2 < 0$,
- $\begin{bmatrix} d_1^T \\ d_2^T \end{bmatrix} H d_3 = 0$, $\nabla q(x_3)^T d_3 < 0$

etc

⊖ · H should be known,
 · need to keep all d_k.

b) there is much better way.

Ex



Th (Theorem 5, p. 78)

Vectors
$$\boxed{\begin{aligned} d_1 &= -\nabla q(x_1), \\ d_{k+1} &= -\nabla q(x_{k+1}) + \beta_k d_k \\ \beta_k &= \frac{\|\nabla q(x_{k+1})\|^2}{\|\nabla q(x_k)\|^2} \end{aligned}}$$

are H-conjugate.

3.5.3. Conjugate gradient method.

Recall Newton: $x_{k+1} = x_k - H^{-1}\nabla f(x_k)$,

i.e. we repeatedly minimized the function f as if it were quadratic.

Let's do the same here: define

quadratic_step(x)

function x_{out} = quadratic_step (x_{in});

$y_1 = x_{in}$;

$d_1 = -\nabla f(x_{in})$;

for $k = 1$ to n

$$(*) \begin{cases} \lambda_k = \text{minimize } f(y_k + \lambda d_k); \\ y_{k+1} = y_k + \lambda_k d_k; \end{cases}$$

$$(**) \begin{cases} \beta_k = \dfrac{\|\nabla f(y_{k+1})\|^2}{\|\nabla f(y_k)\|^2}; \\ d_{k+1} = -\nabla f(y_{k+1}) + \beta_k d_k; \end{cases}$$

end

$x_{out} = y_{n+1}$;

Then x_{k+1} = quadratic_step(x_k)
(Fletcher - Reeves algorithm)

<u>Remark</u>: (*) is a line search.

(**) is the update rule for d_k.

- if we replace (**) with

$$d_{k+1} = -D_{k+1} \nabla f(y_{k+1})$$

where the matrix D_k updates as e.g. in (30), p. 82 (alt. p. 89) then we get a quasi-Newton method (more robust, but needs more information to be kept).

<u>Remark</u>: for <u>quadratic</u> functions $D_{n+1} = H^{-1}$ and d_k are H-conjugate directions.