Lecture 17-18
Primal methods

Solmaz Kia
Mechanical and Aerospace Eng. Dept.,
University of California Irvine

Consult: Section 2.3 of Ref[1] and Sections 12.1,12.2 and 12.4 of Ref[2]

Primal Methods for constraint optimization

We consider the problem

$$\min f(x)$$
$$g(x) \leq 0$$
$$h(x) = 0$$

* Penalty method

* Augmented Lagrangian method (method of multipliers)

Primal-Dual method

* primal method

* Dual method

By a *primal method* we mean a search method that works by searching through the feasible region.

First Order Necessary Condition for Optimality: x^* is a local minimizer then

$$\nabla f(x^*)^\top \Delta x \geqslant 0, \quad \text{for } \Delta x \in V(x^*)$$

- Set of first order feasible variations at x

$$V(x) = \{d \in \mathbb{R}^n \mid \nabla h_i(x)^\top d = 0,\ \nabla g_j(x)^\top d \leqslant 0, \quad j \in A(x^*)\}$$

- Active inequality constraints at x

$$A(x) = \{j \in \{1, \cdots, r\} \mid g_j(x) = 0\}$$

Primal Methods for constraint optimization

We consider the problem

$$\min f(\mathbf{x})$$
$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

By a *primal method* we mean a search method that works by searching through the feasible region.

you start with a feasible x_0 : $\{x_k\}$ are generated in a way that $x_k \in \Omega$

Advantages

- If the process is terminated before reaching the solution, the current point is feasible. ✓ ← Any time control
- It can often be guaranteed that if the sequence of points converges, then the limit is a local constrained minimum. ✓
- Most of the primal methods do not rely on special structure, such as convexity.

Disadvantages

- Needs a feasible starting point.
- It can be computationally hard to remain in the feasible region.

Feasible Direction Methods

The idea of feasible direction methods is the same as with unconstrained problems:

feasible direction

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \in \Omega$$

$x_k + d_k \in \Omega$

where $\mathbf{d}_k$ is a feasible direction at $\mathbf{x}_k$, and $\alpha_k \geq 0$.

α_k is chosen to minimize f with the restriction that the point $\mathbf{x}_{k+1}$, and the line segment joining $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ be feasible.

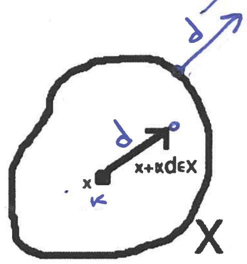
OPT: $x^* = \underset{x \in \mathbb{R}^n}{\text{argmin}}\ f(x)$

$x \in X$ (X is the set of constraints)

for $X = \mathbb{R}^n$ (problem becomes unconstrained)

$D \in \mathbb{R}^n$ is a **feasible direction** at $x \in X$ for OPT if $(x + \alpha d) \in X$ for $\alpha \in [0, \bar{\alpha}]$

not a feasible direction



$x_{k+1} = x_k + \alpha d_k \in \Omega$

$\exists\ \alpha \in [0, \bar{\alpha}]$

$f(x_k) \geq f(x_k + \alpha_k d_k) \approx f(x_k) + \alpha_k \underbrace{\nabla f(x_k)^T d_k}_{\leq 0}$

A Feasible Direction Method: Simplified Zoutendijk method

Consider the a problem with linear constraints:

unconstrained optimization

d_k: descent direction

$\nabla f(x_k)^T d_k \leq 0$

$$\begin{aligned} &\text{minimize } f(x) \quad \text{s.t.} \\ &\qquad a_i^T x \leqslant b_i, \quad i = 1, \cdots, r \end{aligned}$$

Given a feasible point x_k, let $A(x_k)$ be the set of indices of active constraints, i.e., $a_i^T x_k = b_i$ for $i \in A(x_k)$.
The direction vector d_k is then chosen as the solution of

$x_{k+1} = x_k + d_k \in \Omega, A(x_k)$

start with x_0 in Ω

$\{x_k\} \in \Omega$

$$\begin{aligned} &\underset{d}{\text{minimize}}\ \nabla f(x_k)^T d \quad \text{s.t.} \\ &\qquad a_i^T d \leqslant 0, \quad i \in A(x_k), \\ &\qquad \sum_{j=1}^{n} |d_j| = 1 \end{aligned}$$

$d_k = \arg\min$

linear programming

we have effective solvers for linear programming that will solve for d_k

$x_k \in \Omega \Rightarrow$

$a_i^T x_k \leq b_i \quad i \in A(x_k)$

$a_i^T (x_k + d_k) \leq b_i \quad i \in A(x_k)$

$a_i^T d_k \leq 0$

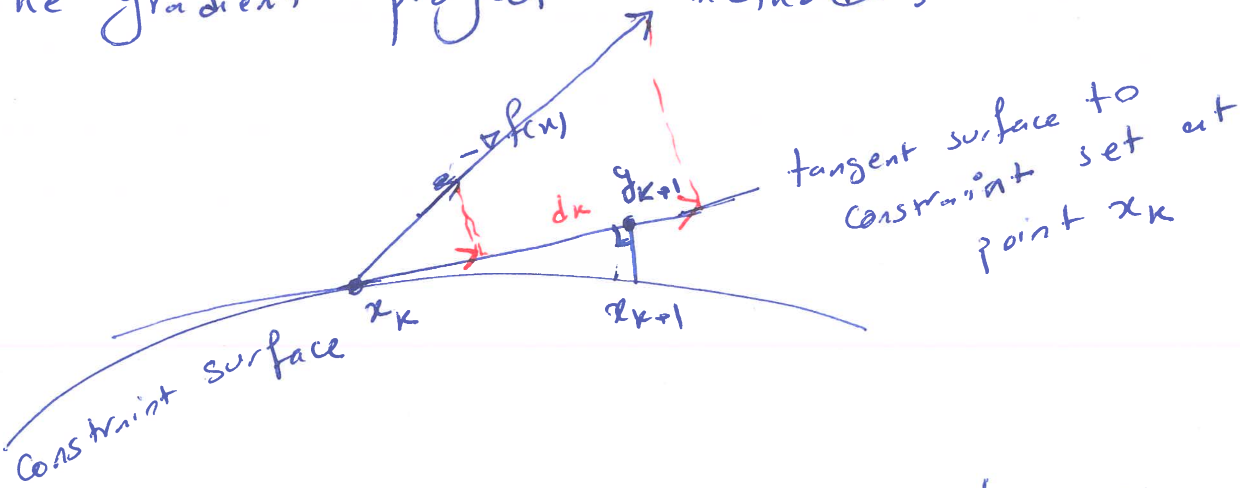
- The last equation (which can be converted to linear constraints) ensures a bounded solution.
- The other constraints assure that $x_k + d_k$ will be feasible for sufficiently small $\alpha_k > 0$.
- The objective function makes d as close to $\nabla f(x_k)$ as possible.

Feasible Direction Methods

There are two major shortcomings of feasible direction methods in this form.

- For general problems there may not exist any feasible directions (see figure). For example in case of nonlinear equality constraints.
- They are also vulnerable to „jamming", or „zigzagging", which we have also seen with unconstrained problems, but here it might converge to a point that is not even a constrained local minimum (the algorithmic map is not closed).

The gradient projection methods:



1- a move is made along the projected negative gradient to a point y_{k+1}

2. a move is made in the direction perpendicular to the tangent plane at the y_{k+1} to a nearby feasible point on the working surface.

projective gradient method

Consider
$$\min f(x)$$
$$a_i^T x \leq b_i \quad i \in J_1 = \{1, \dots, r\}$$
$$a_i^T x = b_i \quad i \in J_2 = \{1, \dots, m\}$$

start with $x_0 \in \Omega$, $\{x_k\}$, $x_k \in \Omega$

At a given feasible point x : $\exists q$ active constraints $a_i^T x = b_i$

and some are inactive $a_i^T x < b_i$

* working set: set $W(x)$ to be the set of active constraints (including equalities)

At the feasible point x we want to move in direction d (feasible direction) such that
$$\nabla f(x)^T d < 0$$
↳ movement in direction d to decrease the function f

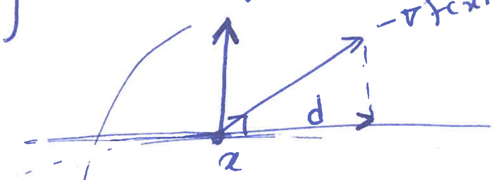
we want direction d to satisfy
$$a_i^T d = 0, \quad i \in W(x)$$
$$x' = x + d \Rightarrow \cancel{a_i^T (x+d) = b_i} \quad i \in W(x)$$
$$\Rightarrow a_i^T b_i = 0 \quad i \in W(x)$$

$a_i^T x = b_i \quad i \in W(x) \qquad |W(x)| = q$

$A_q \in R^{q \times n}$

$A_q = \begin{bmatrix} a_i^T \\ \vdots \\ a_j^T \end{bmatrix} \Rightarrow h(x) = A_q x = b \qquad \text{rank}(A_q) = q$

$b = \begin{bmatrix} b_i \\ \vdots \\ b_j \end{bmatrix}$



$\nabla h(x)^T y = 0$

$M = \{ y \in R^n \mid A_q y = 0 \}$

$A_q^T \Leftarrow \nabla h(x) \, \lambda \qquad \lambda \in R^q$

$\nabla h(x) = \{ \nabla h_i(x) \}_{i \in W(x)} \qquad \sum_{i \in W(x)} \nabla h_i(x) \lambda_i$

$\nabla f(x_k) = g(x_k)$

$-g(x_k) = d_k + A_q^T \lambda_k$

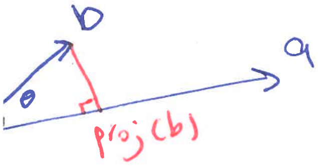
$A_q d_k = 0 \leftarrow$ so $d_k \in M$ (tangent space)

$-A_q g(x_k) = + \cancel{A_q d_k}^{\,0} + A_q A_q^T \lambda_k$

$\lambda_k = -\underbrace{(A_q A_q^T)^{-1}}_{\text{the inverse exists because}} A_q g_k$

$\text{rank}(A_q) = q \qquad A_q \in R^{q \times n}$

$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\theta$



$\text{proj}_{\vec{a}} \vec{b} = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}|} \left(\frac{\vec{a}}{|\vec{a}|} \right) = \frac{\vec{a} \cdot \vec{b}}{\vec{a} \cdot \vec{a}} \vec{a}$

$d_k = -\underbrace{[I - A_q^T (A_q A_q^T)^{-1} A_q]}_{P_k} g_k = -P_k g_k$

$P_k \Leftarrow$ projection operator

* d_k is a descent direction because it satisfies

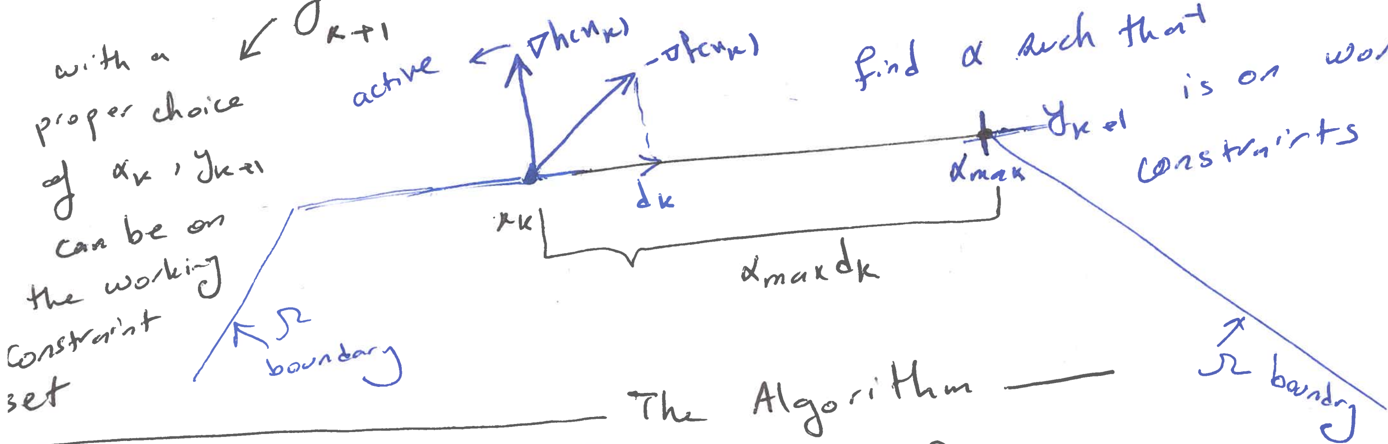
$$\nabla f(x_k)^T d_k \leqslant 0 \quad \text{or} \quad g(x_k)^T d_k \leqslant 0$$

proof

$$g(x_k)^T d_k = -(d_k + A_q^T \lambda_k)^T d_k =$$

$$-d_k^T d_k + \lambda_k^T \underbrace{\cancel{A_q d_k}}_{0} = -d_k^T d_k \leqslant 0$$

$$y_{k+1} = x_k + \alpha_k d_k$$



The Algorithm

1. Start with x_0 feasible; $x_0 \in \Omega$

2. Find $W(x_k)$ and form A_q

3. Calculate $P = I - A_q^T (A_q A_q^T)^{-1} A_q$ and

$$d_k = -P g(x_k)$$

4. If $d \neq 0$, find α_1 and α_2 achieving, respectively

$$\alpha_1 = \arg\max \{\alpha : x + \alpha d_k \text{ is feasible}\}$$

on the working constraint set

$$\alpha_2 = \arg\min_{\alpha \in [0, \alpha_1]} f(x + \alpha d_k)$$

set x_k to $x_k + \alpha_2 d_k$ and go to step 2.

(see next page)

ii) If $d_k = 0$; Compute $\lambda_k = -(A_q A_q^T)^{-1} A_q g_k$

a) If $(\lambda_k)_j \geq 0$ for all j corresponding to active inequality constraints in $W(x_k)$ stop ; x_k satisfies the KKT condition.

(b) otherwise delete the row corresponding to the inequality constraint with most negative $(\lambda_k)_i$ from A_q and from $W(x_k)$, repeat from step (2) with the new A_q and $W(x_k)$

If $d_k = 0$ $\quad -g_k = d_k + A_q^T \lambda_k \Longrightarrow$

$$-d_k = g_k + A_q^T \lambda_k = 0$$

potentially you have satisfied KKT condition if $(\lambda_k)_i$ components corresponding to inequality constraints are nonnegative

$$\left\{ \begin{array}{l} g_k + A_q^T \lambda_k = 0 \\ \nabla f(x_k) + \nabla h(x_k) \lambda_k = 0 \end{array} \right.$$

$\Longleftarrow$ you have satisfied the first order optimality condition (?)

$$\lambda_k = -(A_q A_q^T)^{-1} A_q g_k$$

↳ check the sign of $(\lambda_k)_{i \in \{1,\dots,q\}}$

if all the $(\lambda_k)_i$ corresponding to the inequality constraints are nonnegative (≥ 0)

$x_{k+1} = x_k$ is a stationary point

if one or more of $(\lambda_k)_i$ corresponding to the inequality constraints are negative then drop the inequality constraint corresponding to the most negative $(\lambda_k)_i$ (for inequalities)

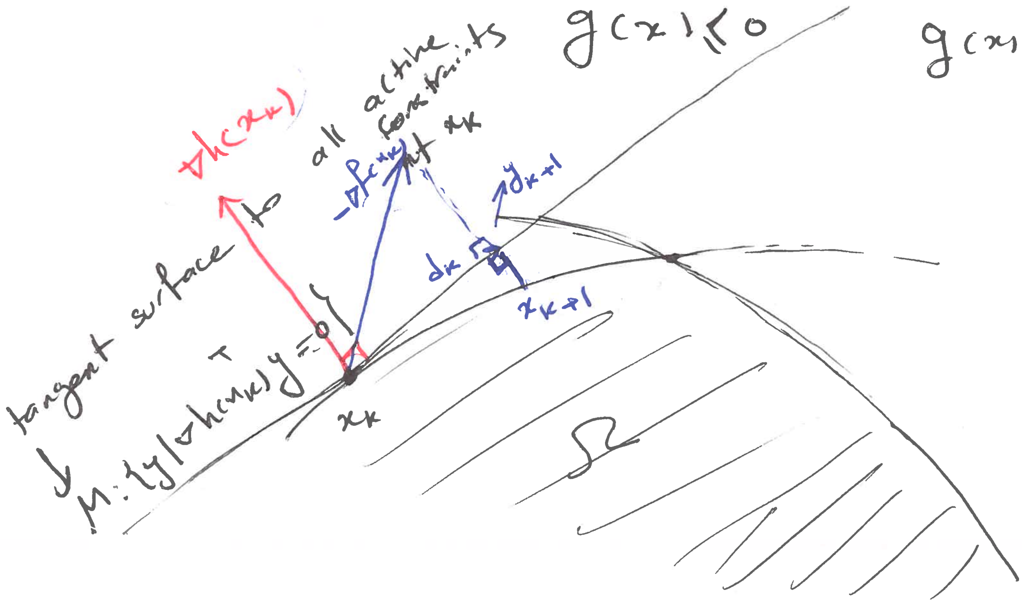
form the new $W(x_k)$ and repeat the process.

* gradient projection (Projective gradient method)
(non-linear constraints)

$$\min f(x)$$

$$h(x) = 0 \qquad h(x) = \begin{bmatrix} h_1 \\ \vdots \\ h_m \end{bmatrix}$$

$$g(x) \leq 0 \qquad g(x) = \begin{bmatrix} g_1 \\ \vdots \\ g_p \end{bmatrix}$$



For notational simplicity Lets lump all active inequality constraints in $h(x_k) = 0 \leftarrow \begin{bmatrix} h \\ [g_i]_{i \in A(x_k)} \end{bmatrix}$

assumption x_k is regular

we choose d_k as projection of $-\nabla f(x_k)$ on the tangent plane M:

$$P_k = I - \nabla h(x_k)^T \left[\nabla h(x_k) \nabla h(x_k)^T\right]^{-1} \nabla h(x_k)$$

$$d_k = -P_k g(x_k)$$

$$y^* = y + \nabla h(x_k)\, w$$

$$\Downarrow$$

$$h(y^*) = 0$$

both equality and active inequalities at x_k

$$h(y + \nabla h(x)\, w) \simeq h(y) + \nabla h(x_k)^T \nabla h(x_k)\, w \simeq 0$$

$$w = -(\nabla h(x_k)^T \nabla h(x_k))^{-1} h(y)$$

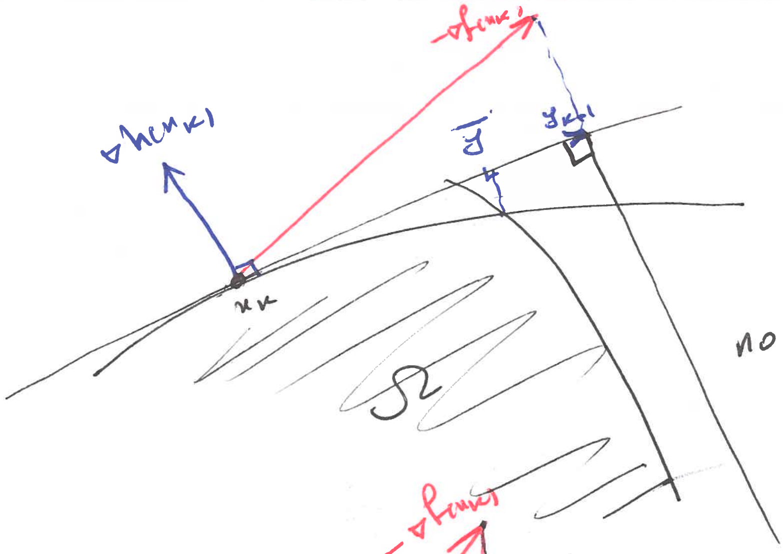
because we used the first order approximation

$$y^* = y - \nabla h(x_k) (h(x_k)^T \nabla h(x_k))^{-1} h(y)$$

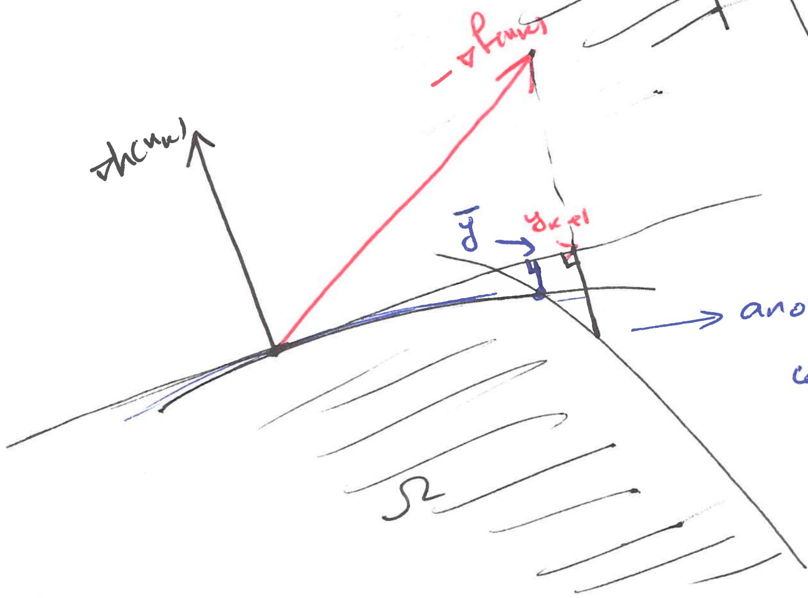
is not going to necessarily satisfy $h(y^*) = 0$

we set $y = y^*$ and repeat the process until

$$\| h(y^*) \| \leqslant \varepsilon$$



no intersection with feasible zone!



another set of constraints are active

the choice of y_{k+1} should be safe guarded.

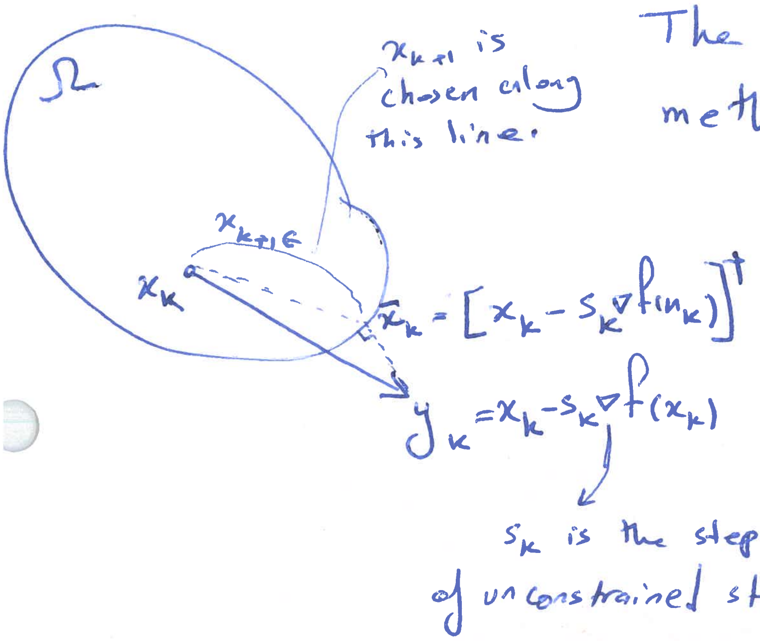
Gradient Projection Method (when Ω is convex)

The basic idea in gradient projection methods is that first we use ideas from what we learned in unconstrained optimization about descent direction to come up with a direction to move that increases the cost. If the move in this direction ends up at a point outside the feasible region, the we project back to the feasible region and use this projected point to come up with a descent direction in the feasible zone.

There are many variations of gradient projection method. In the following we discuss a method to solve

$$\min f(x) \quad \text{s.t.} \quad x \in \Omega$$

where Ω is the feasible set of the optimization problem. Here we assume that Ω is convex. (for more details check Section 2.3 of Ref [1]).



The simplest gradient projection method is a feasible direction method of the form

$$x_{k+1} = x_k + \alpha_k (\bar{x}_k - x_k)$$

where

$$\bar{x}_k = [x_k - s_k \nabla f(x_k)]^+$$

and $\alpha \in (0, 1]$.

* Gradient Projection method :

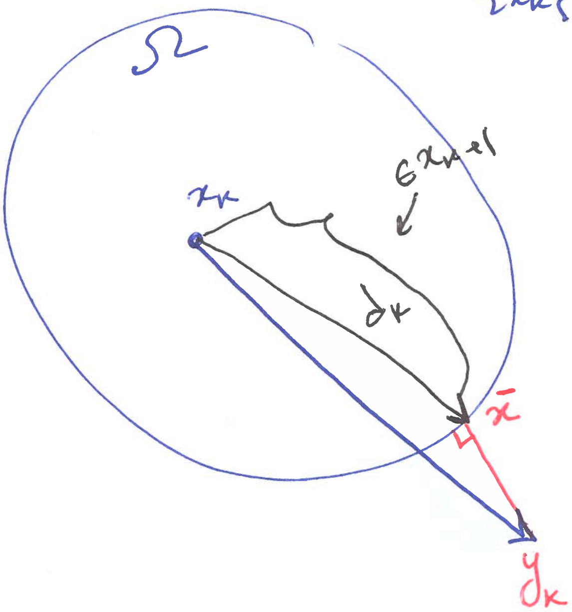
$$\min_{x \in \Omega} f(x)$$

convex

check section 2.3 of Ref [1]

1- start with x_0 inside the feasible set Ω

$\{x_k\}$; $x_k \in \Omega$



2- Use the unconstrained gradient descent method to obtain

$$y_k = x_k - s_k \nabla f(x_k)$$

step size of unconstrained gradient descent

$$\bar{x} = [x_k - s_k \nabla f(x_k)]^+$$

$d_k = \bar{x}_k - x_k$ is a feasible direction

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

$$\boxed{x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k)}$$

$\alpha_k \in [0, 1]$

x_{k+1} is gauranteed to be in Ω (because Ω is convex)

$$\alpha_k = \arg\min_{\alpha \in [0,1]} f(x_k + \alpha d_k)$$

why $f(x_{k+1}) \leq f(x_k)$

Here, $[\cdot]^+$ denotes projection on the set Ω.

$\alpha_k \in (0,1]$ is a stepsize.

To obtain the vector $\bar{x}_k$, we take a step $-s_k \nabla f(x_k)$ along the negative gradient, as in steepest descent. We then project the result $x_k - s_k \nabla f(x_k)$ on Ω, thereby obtaining the feasible vector $\bar{x}_k$. Finally, we take a step along the feasible direction $(\bar{x}_k - x_k)$ using the stepsize α_k.

* Note here that since Ω is a close convex set, therefore $(\bar{x}_k - x_k) \in \Omega$.

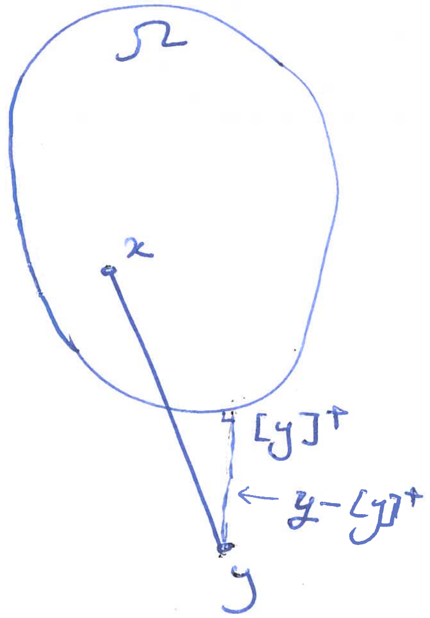
* Note that any $\alpha_k > 1$ will result in $x_{k+1} \notin \Omega$.

Next we study some of the properties of the gradient projection method introduced above. Our study relies on properties of the projection operation discussed in the Projection Theorem below.

The projection operator maps any point $y \notin \Omega$ to the closest point on Ω.

Projection Theorem

Let Ω be a nonempty, closed and convex subset of $\mathbb{R}^n$. Then for any $y \in \mathbb{R}^n$, there exists a unique $x \in \Omega$ denoted by $[y]^+$ such that $[y]^+$ solves the minimization problem



$$\min_{x \in \Omega} \|y - x\|$$

$[y]^+$ is called the projection of y on Ω.

The projection operator has the following properties.

(a) $(y-[y]^+)^T(x-[y]^+) \leq 0$ for $\forall x \in \Omega$

(b) Let $\varphi : \mathbb{R}^n \to \Omega$ be defined by $\varphi(y) = [y]^+$. Then φ is continuous and

$$\|[w]^+ - [v]^+\| \leq \|w - v\| \quad \text{for } w, v \in \mathbb{R}^n.$$

(c) If Ω is a subspace in $\mathbb{R}^n$, $\bar{y} = [y]^+$ if and only if $(y - \bar{y}) \perp \Omega$.

* * * *

In the gradient projection method

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

$$d_k = (\bar{x}_k - x_k)$$

$$\bar{x}_k = [x_k - s_k \nabla f(x_k)]^+$$

Question: Is the direction of the gradient projection method a descent direction?

We need to show $\nabla f(x_k)^T d_k \leq 0$ for $d_k \neq 0$

or

$$\nabla f(x_k)^T (\bar{x}_k - x_k) \leq 0 . \quad (1)$$

we use property (a) of the projection theorem:

$x_k \in \Omega$ $\quad$ $x_k - s_k \nabla f(x_k)$ is projected to Ω to obtain $\bar{x}_k$)

$$((x_k - s_k \nabla f(x_k)) - \bar{x}_k)^T (x_k - \bar{x}_k) \leq 0$$

$$(y - [y]^+)(x - [y]^+) \leq 0$$

$$((x_k - s_k \nabla f(x_k)) - [x_k - s_k \nabla f(x_k)]^+)^T (x_k - [x_k - s_k \nabla f(x_k)]^+) \leq 0$$

$$(-s_k \nabla f(x_k) + (x_k - [x_k - s_k \nabla f(x_k)]^+))^T (x_k - [x_k - s_k \nabla f(x_k)]^+) \leq 0$$

$$-s_k \nabla f(x_k)^T (x_k - \bar{x}_k) + (x_k - [x_k - s_k \nabla f(x_k)]^+)^T (x_k - [x_k - s_k \nabla f(x_k)]^+) \leq 0$$

$$s_k \nabla f(x_k)^T (\bar{x}_k - x_k) \leq -\|\bar{x}_k - x_k\|^2$$

then for any $\bar{x}_k \neq x_k$ we have (1).

jection operator satisfied $(y-[y]^+)^T(x-[y]^+) \leq 0$ for any $\forall x \in \Omega$

$$(\underbrace{x_k - s_k \nabla f(x_k)}_{y} - [x_k - s_k \nabla f(x_k)]^+)^T (x_k - [x_k - s_k \nabla f(x_k)]^+) \leq$$

$$-s_k \nabla f(x_k)^T (x_k - \underbrace{[x_k - s_k \nabla f(x_k)]^+}_{\bar{x}}) + \| x_k - [\underbrace{x_k - s_k \nabla f(x_k)}_{\bar{x}}]^+ \|^2 \leq$$

$$-s_k \nabla f(x_k)^T (\underbrace{x_k - \bar{x}}_{-d_k}) \leq -\| x_k - \bar{x} \|^2$$

$$s_k \nabla f(x_k)^T d_k \leq -\| x_k - \bar{x} \|^2$$

$$\Downarrow$$

$$\nabla f(x_k)^T d_k \leq 0$$

$$\Downarrow$$

so d_k is a descent direction

$\exists \, \alpha_k \in [0,1]$ $\quad f(x_{k+1}) \leq f(x_k)$

Note that we have $x^* = [x^* - s\nabla f(x^*)]^+$ for all $s>0$ if and only if x^* is stationary point. Thus, the gradient projection stops if and only if it encounters a stationary point

Lets say at step k we have $[x_k - s_k \nabla f(x_k)]^+ = x_k$

then using property (a) of gradient projection we can write

$(y - [y]^+)^T (x$

$$\left((x_k - s_k \nabla f(x_k)) - [x_k - s_k \nabla f(x_k)]^+\right)^T (x - [x_k - s_k \nabla f(x_k)]^+) \leq 0$$

for any $x \in \Omega$

$[x_k - s_k \nabla f(x_k)]^+ = x_k$

$$(x_k - s_k \nabla f(x_k) - x_k)^T (x - x_k) \leq 0$$

$$-s_k \nabla f(x_k)^T (x - x_k) \leq 0 \implies$$

$$\therefore \boxed{\nabla f(x_k)^T (x - x_k) \geq 0 \quad \forall x \in \Omega}$$

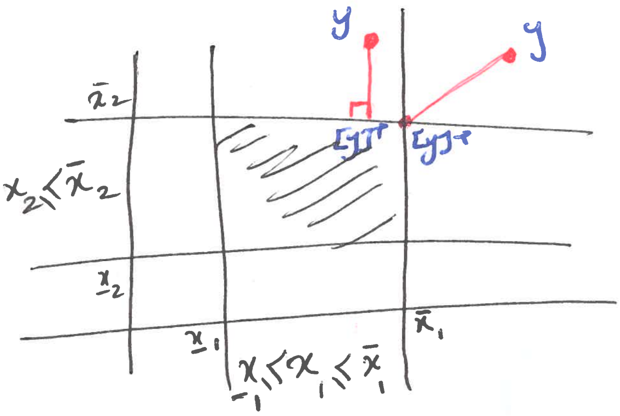
Recall the first order necessary condition for optimality.

Examples of projection operator

$$\begin{cases} \min_{x \in R^n} f(x) \quad s.t. \\ \underline{x}_i \leq x_i \leq \bar{x}_i \end{cases} \longleftarrow \text{box inequality}$$

$$y_k \Rightarrow [y_k]^+ = \begin{cases} (y_k)_i & \underline{x}_i \leq (y_k)_i \leq \bar{x}_i \\ \underline{x}_i & (y_k)_i < \underline{x}_i \\ \bar{x}_i & (y_k)_i > \bar{x}_i \end{cases}$$

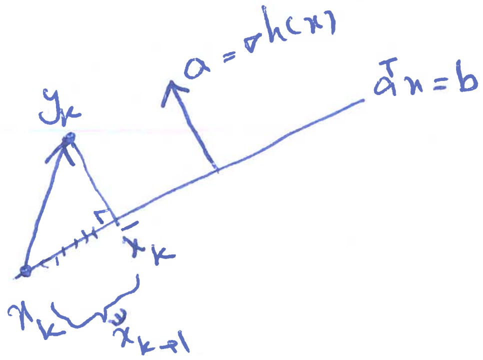
$$\begin{matrix} \underline{x}_1 \leq \\ \\ \underline{x}_n \leq \end{matrix} \; y_k = \begin{bmatrix} (y_k)_1 \\ \vdots \\ (y_k)_n \end{bmatrix} \begin{matrix} \leq \bar{x}_1 \\ \\ \leq \bar{x}_n \end{matrix}$$



Examples:

Consider $\min f(x)$

$a^T x = b \in R$



$$y_k = x_k - s_k \nabla f(x_k)$$

$$[y_k]^+ = \bar{x}_k = y_k - w a$$

$$\bar{x}_k \in \Omega \Rightarrow a^T(y_k - wa) = b \Rightarrow a^T y_k - w a^T a = b \Rightarrow$$

$$w = \frac{b - a^T y_k}{a^T a} = -\frac{b - a^T(x_k - s_k \nabla f(x_k))}{a^T a}$$

$$w = -\frac{s_k a^T \nabla f(x_k)}{a^T a}$$

$$\Rightarrow [y_k]^+ = \bar{x}_k = x_k - s_k \nabla f(x_k) + \frac{s_k a^T \nabla f(x_k) a}{a^T a} =$$

$$x_k - s_k \left(I - \frac{a^T a}{a^T a}\right) \nabla f(x_k) =$$

$$x_k - s_k \left(I - a^T (a^T a)^{-1} a\right) \nabla f(x_k)$$

Recall that projection of $y_k - x_k$ onto $a^T x = b$ is

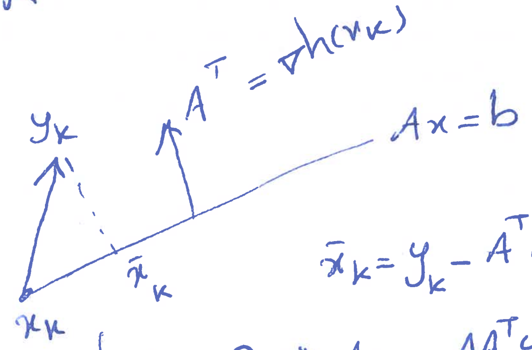
$$\bar{x}_k - x_k = (I - a^T (a^T a)^{-1} a)(y_k - x_k)$$

$$[y_k]^+ = \bar{x}_k = x_k + (I - a^T (a^T a)^{-1} a)(-s_k \nabla f(x_k))$$

$$= x_k - s_k (I - a^T (a^T a)^{-1} a) \nabla f(x_k)$$

Examples $\quad \min f(x)$

$$Ax = b \in \mathbb{R}^m \qquad \operatorname{rank}(A) = m$$



projection of $y_k - x_k$ using projection matrix (see the earlier derivation of projection matrix)

$$\bar{x}_k = y_k - A^T\omega$$

$$\bar{x}_k \in \Omega \Rightarrow Ay_k - AA^T\omega = b$$

$$\cancel{Ax_k} - As_k \nabla f(x_k) - AA^T\omega = \cancel{b}$$

$$\omega = -s_k (AA^T)^{-1} A \nabla f(x_k)$$

$$\bar{x}_k = y_k - A^T\omega \Rightarrow$$

$$\bar{x}_k = x_k - s_k (I - A^T(AA^T)^{-1}A) \nabla f(x_k)$$

$$P = (I - A^T(AA^T)^{-1}A)$$

is

$$\bar{x}_k - x_k = P(y_k - x_k)$$

$$\bar{x}_k - x_k = (I - A^T(AA^T)^{-1}A)(y_k - x_k)$$

$$\bar{x}_k = x_k + (I - A^T(AA^T)^{-1}A)(-s_k \nabla f(x_k))$$

$$\bar{x}_k = [y_k]^+ = x_k - s_k (I - A^T(AA^T)^{-1}A) \nabla f(x_k)$$

Gradient Projection Algorithm:

$$\min_{x \in \Omega} f(x)$$

Ω $\{Ax = b$ Ω $\} Ax \leq b$

Ω $\begin{cases} Ax = b \\ g(x) \leq 0 \end{cases}$ convex

1- Initialize with $x_0 \in \Omega$

2- Implement one step of gradient descent for unconstrained optimization

$$y_k = x_k - s_k \nabla f(x_k)$$

* $\bar{x}_k = [y_k]^+$ where $[\cdot]^+$ denotes the projection onto Ω operator

* stop if $\bar{x}_k = x_k = [x_k - s_k \nabla f(x_k)]^+$; otherwise

* $x_{k+1} = x_k + \alpha_k (\bar{x}_k - x_k)$

$$\alpha_k = \underset{\alpha \in [0,1]}{\arg\min}\, f(x_k + \alpha d_k) \; ; \; d_k = \bar{x}_k - x_k$$

go to step 2.

PRACTICAL AUGMENTED LAGRANGIAN METHODS:
BOUND-CONSTRAINED FORMULATION

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\qquad\qquad h(x) = 0, \qquad l < x < u \end{aligned}$$

$$L_A(x,\lambda^k;c_k) = f(x) + \sum_{i=1}^{m} \lambda_i^k h_i(x) + \frac{c_k}{2}\sum_{i=1}^{m} h_i(x)^2$$

Bounded Gradient Lagrangian method

$$\begin{cases} x_k \leftarrow \operatorname{argmin} L_A(x,\lambda^k;\mu_k) \ \text{ subject to} \\ \qquad l < x < u \end{cases}$$

$$\lambda_i^{k+1} = \lambda_i^k + c_k h_i(x_k)$$

$$c_{k+1} > c_k > 0$$

$$\boxed{\begin{array}{l} \text{minimize} \quad f(x) \\ \qquad\qquad h(x) = 0, \qquad l < x < u \end{array}}$$

(Bound-Constrained Lagrangian Method).

Choose an initial point x_0 and initial multipliers λ^0;

Choose convergence tolerances η_* and ω_*;

Set $C_0 = 10$, $\omega_0 = 1/C_0$, and $\eta_0 = 1/C_0^{0.1}$;

for $k = 0, 1, 2, \ldots$

Find an approximate solution x_k of the subproblem $\min_x \mathcal{L}_A(x,\lambda^k;C_k)$ subject to $l \le x \le u$ such that

$$\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; C_k), l, u\right) \right\| \le \omega_k;$$

if $\|h(x_k)\| \le \eta_k$

(* test for convergence *)

if $\|h(x_k)\| \le \eta_*$ **and** $\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; C_k), l, u\right) \right\| \le \omega_*$

stop with approximate solution x_k;

end (if)

(* update multipliers, tighten tolerances *)

$\lambda^{k+1} = \lambda^k + C_k h(x_k)$;

$C_{k+1} = C_k$;

$\eta_{k+1} = \eta_k / C_{k+1}^{0.9}$;

$\omega_{k+1} = \omega_k / C_{k+1}$;

else

(* increase penalty parameter, tighten tolerances *)

$\lambda^{k+1} = \lambda^k$;

$C_{k+1} = 100 C_k$;

$\eta_{k+1} = 1/C_{k+1}^{0.1}$;

$\omega_{k+1} = 1/C_{k+1}$;

end (if)

end (for)

P here is the projection operator for boxed inequality (check your notes on gradient projection method for further details)

An efficient technique for solving the nonlinear program with bound constraints (for fixed μ and λ) is the (nonlinear) gradient projection method (see your notes from lectures on primal methods)

Remember that gradient projection method stops when the point generated from x_k by the gradient descent algorithm gets projected back on x_k. Check the stopping condition of the gradient projection method in your notes for more details.

If this condition holds, the penalty parameter is not changed for the next iteration because the current value of μ_k is producing an acceptable level of constraint violation. The Lagrange multiplier estimates are updated according to the update formula and the tolerances ω_k and η_k are tightened in advance of the next iteration. If, on the other hand, this condition does not hold, then we increase the penalty parameter to ensure that the next subproblem will place more emphasis on decreasing the constraint violations. The Lagrange multiplier estimates are not updated in this case; the focus is on improving feasibility.

The constants 100, 0.1, 1 appearing here are to some extent arbitrary; other values can be used without compromising theoretical convergence properties.