Optimization with Constraints

Prof. Cesar de Prada
ISA-UVA
prada@autom.uva.es

Outline

- Constraints
- Equality constraints
  - Lagrange multipliers
- More general problems NLP
  - Karush-Kuhn-Tucher conditions (KKT, KTC)
- Quadratic Programming QP
- Wolfe method, SLP
- Penalty functions

Constraints

Most of the practical decision making problems involve several relations among variables or constraints in the values they can get.

✓Some are given by balances, physical laws, etc. and appears as model equations

✓Other are due to the admissible range of the variables

✓Other correspond to operating rules, etc.

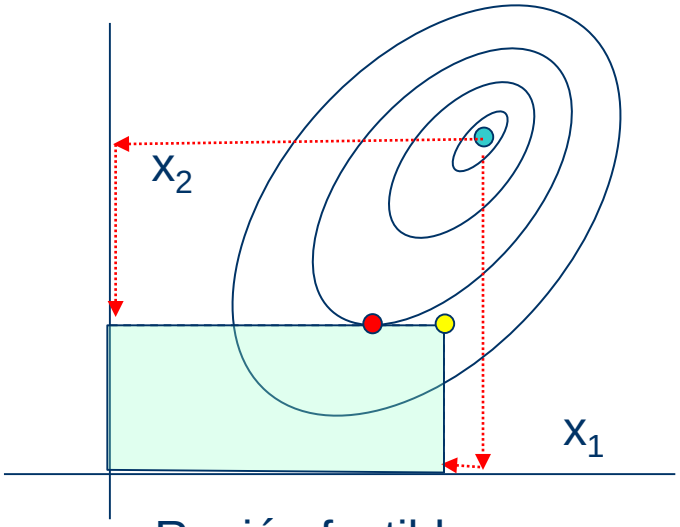
Hence, a typical optimization problem with real variables is formulated as:

$$\min_{x} J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \leq \mathbf{0}$$

Constrained non-linear optimization problem: non-linear programming NLP

Constraints

The existence of constraints reduces the searching space but, at the same time, makes more difficult finding the optimal solution because the zero gradient optimality criterion is no longer true.



Clipping: A policy that first find the unconstraint optimum ● and then clips the solution to the feasible set ● leads to wrong results (the optimum is located at ●, not at ●)

Equality constraints

In many problems some of the variables are linked by nature laws, balances, etc. If there are no other range constraints, the problem is formulated as:

$$\min_{\mathbf{x}} J(\mathbf{x})$$

$$\mathbf{h}(\mathbf{x}) = 0$$

$$\min_{\mathbf{x}} J(x_1, x_2, ..., x_n)$$

$$\left.\begin{array}{l} h_1(x_1, x_2, ..., x_n) = 0 \\ h_2(x_1, x_2, ..., x_n) = 0 \\ .... \\ h_m(x_1, x_2, ..., x_n) = 0 \end{array}\right\} \quad n > m$$

If it is possible to work out m variables as a function of the remaining n-m ones, then they can be substituted in J(x) and the problem is converted in an unconstrained optimization one with n-m variables

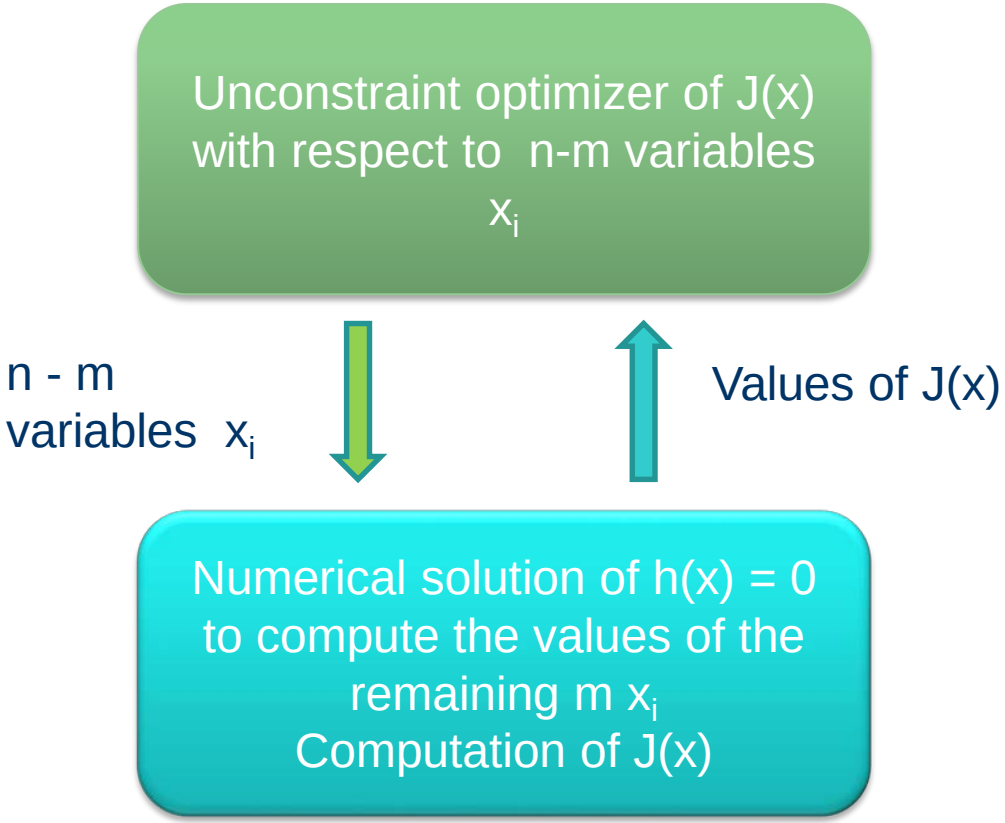
Equality constraints

$$\left.\begin{array}{l}\min\limits_{x_1,x_2}\ (x_1x_2+x_2^2)\\ x_1\log x_2=x_2\end{array}\right\}\ \Rightarrow x_1=\frac{x_2}{\log x_2}\ \Rightarrow \min_{x_2}\frac{x_2^2}{\log x_2}+x_2^2$$

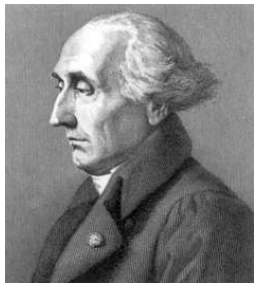
The same happens if starting from a value of $x_1,x_2,\ldots,x_{n-m}$ it is possible to evaluate $J(x_1,x_2,\ldots,x_n)$ making use of the model $h(x) = 0$ and a solver (for instance in a simulation environment). In this case, an unconstraint optimization algorithm applied with $x_1,x_2,\ldots,x_{n-m}$ as decision variables will solve the problem.

Nevertheless, not always is possible to work out n-m variables as functions of the remaining ones and the use of a simulator may imply more computations as the Newton's method, or same variant, must be used.

Sequential solution using a simulator



Lagrange multipliers



1736 –1813

The Lagrange multipliers method provides necessary conditions that the optimum of a equality constrained optimization problem must fulfil. The idea behind is to convert the original problem into another one with some additional variables (the m Lagrange multipliers λ_j) but unconstrained, and such that the x variables of the solution are the same as the ones of the original problem with the constraints h(x) = 0

L(**x**,**λ**) Lagrangian

$$\mathrm{L}(\mathbf{x},\boldsymbol{\lambda}) = \mathrm{J}(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{h}(\mathbf{x}) = \mathrm{J}(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathrm{h}_i(\mathbf{x})$$

$$\left.\begin{array}{l} \min\limits_{\mathbf{x}} J(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) = 0 \end{array}\right\} \Rightarrow \min_{\mathbf{x},\boldsymbol{\lambda}} L(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x},\boldsymbol{\lambda}} J(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{h}(\mathbf{x})$$

Lagrange multipliers

The solution of $\min_{\mathbf{x},\boldsymbol{\lambda}} L(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x},\boldsymbol{\lambda}} J(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{h}(\mathbf{x})$

verifies $\left.\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \mathbf{x}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*} = \mathbf{0}, \quad \left.\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*} = \mathbf{0} \quad \Rightarrow \quad \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$

hence, it satisfies the constraints h(x)=0 and minimizes J(x), so it solves the EC problem

$$\left.\begin{array}{l} \min_{\mathbf{x}} J(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) = 0 \end{array}\right\}$$

Notice that for all x such that h(x)=0, it happens:

$$\min_{\mathbf{x},\boldsymbol{\lambda}} L(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x}} J(\mathbf{x})$$

Conversely, if x^* is optimum for the original problem, it minimizes $J(x^*)$ and complies with $h(x^*) = 0$, hence, it must also minimize the Lagrangian L

Lagrange multipliers

$$\min_{\mathbf{x},\boldsymbol{\lambda}} L(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x},\boldsymbol{\lambda}} J(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{h}(\mathbf{x})$$

L(**x**,λ) Lagrangian

The solution of the EC optimization problem can be found solving:

$$\left.\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \mathbf{x}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*} = \mathbf{0}, \qquad \left.\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*} = \mathbf{0} \quad \Rightarrow \quad \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$$

The Newton- Raphson method can be used to solve this set of algebraic equations. Once the solution x^*,λ^* has been found, it is necessary to check, using the Hessian, that it corresponds to a minimum of L with respect to x

Constraint qualification

Analytical solution of the Lagrangian optimum:

$$\frac{\partial J(\mathbf{x}^*)}{\partial \mathbf{x}} + \boldsymbol{\lambda}^{*'} \frac{\partial \mathbf{h}(\mathbf{x}^*)}{\partial \mathbf{x}} = \mathbf{0} \qquad \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$$

$$\left[\frac{\partial J(\mathbf{x})}{\partial x_1} \quad \frac{\partial J(\mathbf{x})}{\partial x_2} \quad \cdots \quad \frac{\partial J(\mathbf{x})}{\partial x_n}\right] + \left[\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_m\right] \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial x_1} & \frac{\partial h_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial h_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial h_2(\mathbf{x})}{\partial x_1} & \frac{\partial h_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial h_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_m(\mathbf{x})}{\partial x_1} & \frac{\partial h_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial h_m(\mathbf{x})}{\partial x_n} \end{bmatrix} = \mathbf{0}$$

Constraint qualification

$$\frac{\partial J(\mathbf{x})}{\partial x_1} + \lambda_1 \frac{\partial h_1(\mathbf{x})}{\partial x_1} + \lambda_2 \frac{\partial h_2(\mathbf{x})}{\partial x_1} + \ldots + \lambda_m \frac{\partial h_m(\mathbf{x})}{\partial x_1} = 0$$

$$\frac{\partial J(\mathbf{x})}{\partial x_2} + \lambda_1 \frac{\partial h_1(\mathbf{x})}{\partial x_2} + \lambda_2 \frac{\partial h_2(\mathbf{x})}{\partial x_2} + \ldots + \lambda_m \frac{\partial h_m(\mathbf{x})}{\partial x_2} = 0$$

.....

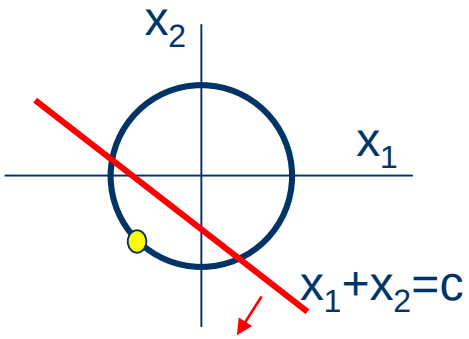
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

n + m equations and n + m unknowns

In order to solve the problem the gradients $\nabla_x h_j$ must be linearly independent, condition that is known as constraint qualification. If not, there would be only n + m -1 unknowns

Example 1

$$\left.\begin{array}{l}\min\limits_{x_1,x_2} x_1 + x_2 \\ x_1^2 + x_2^2 = 4\end{array}\right\}$$



The qualification constraint is fulfilled as there is only one

$$\min_{x_1,x_2,\lambda} L(x_1,x_2,\lambda) = \min_{x_1,x_2,\lambda} x_1 + x_2 + \lambda(x_1^2 + x_2^2 - 4)$$

$$\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial x_1} = 1 + 2\lambda x_1 = 0 \quad \frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial x_2} = 1 + 2\lambda x_2 = 0 \quad \frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \lambda} = x_1^2 + x_2^2 - 4 = 0$$

$$\left.\begin{array}{l}1 + 2\lambda x_1 = 0 \\ 1 + 2\lambda x_2 = 0 \\ x_1^2 + x_2^2 - 4 = 0\end{array}\right\} \quad \left.\begin{array}{l}x_1 = -1/2\lambda \\ x_2 = -1/2\lambda \\ 1/4\lambda^2 + 1/4\lambda^2 = 4\end{array}\right\} \quad \left.\begin{array}{l}\lambda^* = \pm 1/(2\sqrt{2}) \\ x_1 = \mp\sqrt{2} \\ x_2 = \mp\sqrt{2}\end{array}\right.$$

Example 1

$$\min_{x_1,x_2,\lambda} L(x_1,x_2,\lambda) = \min_{x_1,x_2,\lambda} x_1 + x_2 + \lambda(x_1^2 + x_2^2 - 4)$$

$$\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial x_1} = 1 + 2\lambda x_1 \qquad \frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial x_2} = 1 + 2\lambda x_2 \qquad \frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial \lambda} = x_1^2 + x_2^2 - 4$$

$$\mathbf{H}_x(\mathbf{x},\boldsymbol{\lambda}) = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 x_2} \\ \frac{\partial^2 L}{\partial x_2 x_1} & \frac{\partial^2 L}{\partial x_2^2} \end{bmatrix}_{\mathbf{x}^*,\boldsymbol{\lambda}^*} = \begin{bmatrix} 2\lambda & 0 \\ 0 & 2\lambda \end{bmatrix}_{\mathbf{x}^*,\boldsymbol{\lambda}^*} \Rightarrow \begin{bmatrix} \sqrt{2}/2 & 0 \\ 0 & \sqrt{2}/2 \end{bmatrix} \text{PD}$$

$$\begin{bmatrix} -\sqrt{2}/2 & 0 \\ 0 & -\sqrt{2}/2 \end{bmatrix} \text{ND}$$

It suffices that the minimum be with respect to x

This corresponds to $\lambda^*=1/(2\sqrt{2})$: $x_1^* = -\sqrt{2}$, $x_2^* = -\sqrt{2}$

Economic interpretation of the Lagrange multipliers / Sensitivities

$$\left.\begin{array}{l}\min\limits_{\mathbf{x}} J(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) = \mathbf{b}\end{array}\right\} \Rightarrow \min_{\mathbf{x},\boldsymbol{\lambda}} L(\mathbf{x},\boldsymbol{\lambda}) = \min_{\mathbf{x},\boldsymbol{\lambda}} J(\mathbf{x}) + \boldsymbol{\lambda}'\left[\mathbf{h}(\mathbf{x}) - \mathbf{b}\right] \qquad L(\mathbf{x}^*,\boldsymbol{\lambda}^*) = J(\mathbf{x}^*)$$

$0 = h(x) - b$

How much does $J(x^*)$ change if b changes in one unit?

$$\left.\begin{array}{l}\dfrac{\partial \mathrm{J}(\mathbf{x}^*)}{\partial \mathbf{b}} = \dfrac{\partial \mathrm{J}(\mathbf{x}^*)}{\partial \mathbf{x}^*}\dfrac{\partial \mathbf{x}^*}{\partial \mathbf{b}} \\ \mathbf{0} = \dfrac{\partial(\mathbf{h}(\mathbf{x}^*) - \mathbf{b})}{\partial \mathbf{b}} = \dfrac{\partial \mathbf{h}(\mathbf{x}^*)}{\partial \mathbf{x}^*}\dfrac{\partial \mathbf{x}^*}{\partial \mathbf{b}} - \mathbf{I}\end{array}\right\}$$

← Multiplying by $\lambda^{*\prime}$ and adding:

$$\frac{\partial \mathrm{J}(\mathbf{x}^*)}{\partial \mathbf{b}} = \left[\frac{\partial \mathrm{J}(\mathbf{x}^*)}{\partial \mathbf{x}^*} + \boldsymbol{\lambda}^{*\prime}\frac{\partial \mathbf{h}(\mathbf{x}^*)}{\partial \mathbf{x}^*}\right]\frac{\partial \mathbf{x}^*}{\partial \mathbf{b}} - \boldsymbol{\lambda}^{*\prime} = \left[\frac{\partial \mathrm{L}(\mathbf{x}^*,\boldsymbol{\lambda}^*)}{\partial \mathbf{x}^*}\right]\frac{\partial \mathbf{x}^*}{\partial \mathbf{b}} - \boldsymbol{\lambda}^{*\prime} = -\boldsymbol{\lambda}^{*\prime}$$

The optimal value of the Lagrange multipliers (shadow prices) give the (opposite) sensitivities of J^* with respect to the constraints b

Lagrange multipliers/ Shadow prices of LP

Linear programming, LP:

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$
$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq 0$$

Non-linear Programming with equality constraints

$$\left.\begin{array}{l} \min_{\mathbf{x}} J(\mathbf{x}) \\ \mathbf{h}(\mathbf{x}) = \mathbf{b} \end{array}\right\}$$

How much does J(x*) change if b changes in one unit?

The answer (max) using the duality theory of LP is given by the solution z, named as shadow prices

$$\frac{\partial J^*}{\partial \mathbf{b}} = \mathbf{z}^*$$

The answrer (min) using the Lagrangian formulation is given by the negative Lagrange multipliers -λ,

$$\frac{\partial J^*}{\partial \mathbf{b}} = -\boldsymbol{\lambda}^{*\prime}$$

Lagrange multipliers = Shadow prices

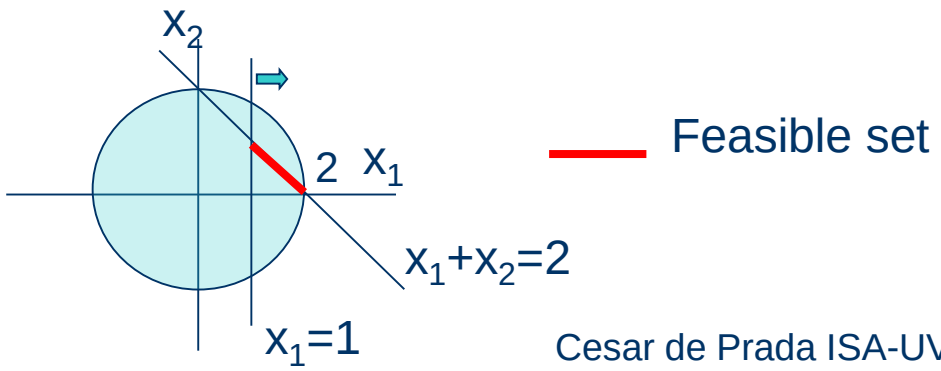
Non linear Programming, NLP

$$\min_{x} J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \leq \mathbf{0}$$

A more general optimization problem includes both types of constraints, equalities and inequalities, on the real decision variables x, which it is known as NLP.

Assuming that functions J, **g** and **h** are continuously differentiable, it is possible to formulate necessary (and in certain cases sufficient) conditions that must be fulfilled by the optimum point x*. These are known as the Karush-Kunt-Tucker (KKT) conditions. (Or KTC, Kunt-Tacker Conditions)

$$\left.\begin{array}{l} \min\limits_{x_1,x_2} x_1^2 - x_2 \\ x_1 + x_2 = 2 \\ x_1^2 + x_1^2 \leq 4 \\ x_1 - 1 \geq 0 \end{array}\right\}$$



Karush(1939) Chicago

Kuhn,Tucker (1951) Berkeley

KKT Optimality conditions

$$\min_x J(\mathbf{x})$$

$$\mathbf{h(x)} = \mathbf{0}$$

$$\mathbf{g(x)} \leq \mathbf{0}$$

The key idea behind KKT conditions is linked to the Lagrangian, considering that, if an inequality constraint is active at **x***, then it can be treated as an equality one, assigning it a Lagrange multiplier μ, while, if it is not active, then it can be ignored so that its Lagrange multiplier μ must be zero. In this way, either μ or g must be zero.

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\mathbf{x}) + \sum_j \lambda_j h_j(\mathbf{x}) + \sum_i \mu_i g_i(\mathbf{x}) \qquad \mu_i g_i(x) = 0$$

On the other hand, if we increase the right hand side of $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$, then the feasible region increases, so that J(**x**) could get a lower value. Hence, the sensitivity of J to **b**, given by $-\boldsymbol{\mu}$, must be negative, that is, $\boldsymbol{\mu} \geq 0$.

KKT Optimality conditions

$$\min_x J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \le \mathbf{0}$$

In this way, the optimal solution of the NLP problem must fulfil the optimality conditions of the Lagrangian L(**x**,λ,μ), plus the additional conditions that guarantee a minimum.

With continuously differentiable functions J, **h** and **g**, the necessary optimality conditions with respect to x are:

$$\left.\begin{aligned}
&\frac{\partial \mathrm{J}}{\partial \mathbf{x}} + \boldsymbol{\lambda}' \frac{\partial \mathbf{h}}{\partial \mathbf{x}} + \boldsymbol{\mu}' \frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \mathbf{0} \\
&\mathbf{h(x)} = \mathbf{0} \\
&\mathbf{g(x)} \le \mathbf{0} \\
&\boldsymbol{\mu}'\mathbf{g(x)} = 0 \\
&\boldsymbol{\mu} \ge \mathbf{0}
\end{aligned}\right\}$$

$$\nabla_x \mathrm{J}(\mathbf{x}) + \sum_j \lambda_j \nabla_x \mathrm{h}_j(\mathbf{x}) + \sum_i \mu_i \nabla_x \mathrm{g}_i(\mathbf{x}) = 0$$
$$\mathrm{h}_j(\mathbf{x}) = 0$$
$$\mathrm{g}_i(\mathbf{x}) \le 0$$
$$\mu_i \mathrm{g}_i(\mathbf{x}) = 0$$
$$\mu_i \ge 0$$

Solving this set of equations one can find a possible optimum x*. Notice that the gradients are required.

KKT optimality conditions, vector form

$$\left.\begin{array}{l} \sum_i \mu_i g_i(\mathbf{x}) = \boldsymbol{\mu}'\mathbf{g}(\mathbf{x}) = 0 \\ \mu_i \geq 0 \\ g_i(\mathbf{x}) \leq 0 \end{array}\right\} \Leftrightarrow \mu_i g_i(\mathbf{x}) = 0$$

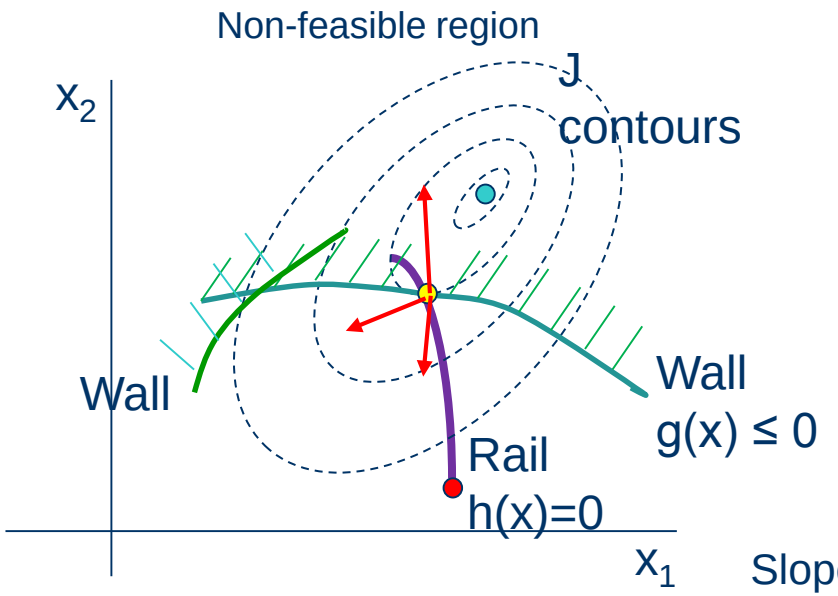
$$\left.\begin{array}{l} \dfrac{\partial J}{\partial \mathbf{x}} + \boldsymbol{\lambda}' \dfrac{\partial \mathbf{h}}{\partial \mathbf{x}} + \boldsymbol{\mu}' \dfrac{\partial \mathbf{g}}{\partial \mathbf{x}} = \mathbf{0} \\ \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ \boldsymbol{\mu}'\mathbf{g}(\mathbf{x}) = 0 \\ \boldsymbol{\mu} \geq \mathbf{0} \end{array}\right\}$$

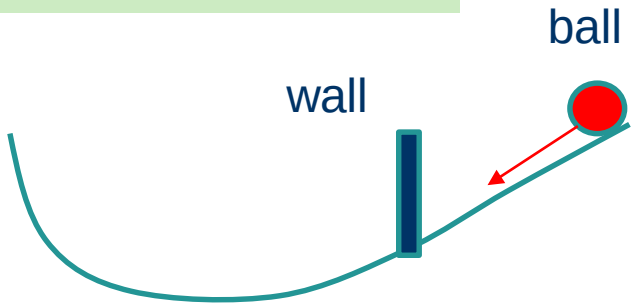
Notice that a linear combination of positive or zero numbers with positive or zero coefficients can only be equal to zero if all terms are zero

$$F = z(x)'y(x) = \sum_i z_i y_i$$

$$\nabla_x F = \sum_i z_i \nabla_x y_i + (\nabla_x z_i) y_i = z'\nabla_x y + y'\nabla_x z$$

KKT





Equilibrium of forces

$$\nabla J + \lambda \nabla h + \mu \nabla g = 0$$

Slope of J

Orthogonal to the rail

Orthogonal to the wall

Only the forces associated to active walls should be considered

Constraint Qualifications

Notice that, for the active constraints at the optimum, one can write:

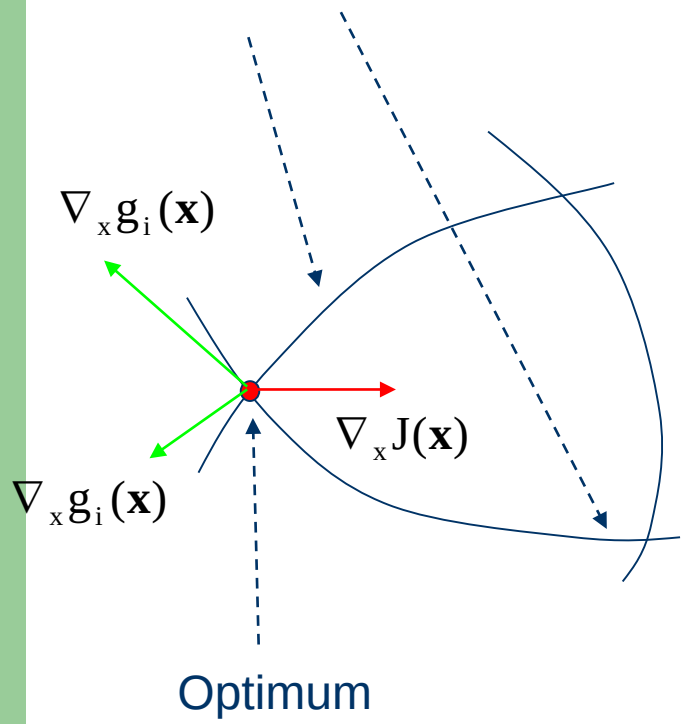
$$\begin{bmatrix}\nabla_x h(\mathbf{x}) & \nabla_x g(\mathbf{x})\end{bmatrix}\begin{bmatrix}\lambda \\ \mu\end{bmatrix} = -\nabla_x J(\mathbf{x})$$

So that $\nabla_x g_i$ y $\nabla_x h_j$ must be linearly independent in order to have a unique solution of this systems of equations. This condition is known as the constraint qualification and can be difficult to verify. Nevertheless, it is automatically verified:

✓When the constraints are linear

✓When the equality constraints are linear and the inequalities are convex and, in addition, there exist a feasible point in the interior of the feasible region created by the inequalities.

At the optimum



In a problem with inequality constraints:

$$\nabla_x J(\mathbf{x}) = -\sum_i \mu_i \nabla_x g_i(\mathbf{x})$$

At the optimum x*, for all active constraints it happens $\mu > 0$, then the gradient of J, that is a linear combination in μ of the gradients of $g_{j\ (active)}$, must be in the opposite half-plane

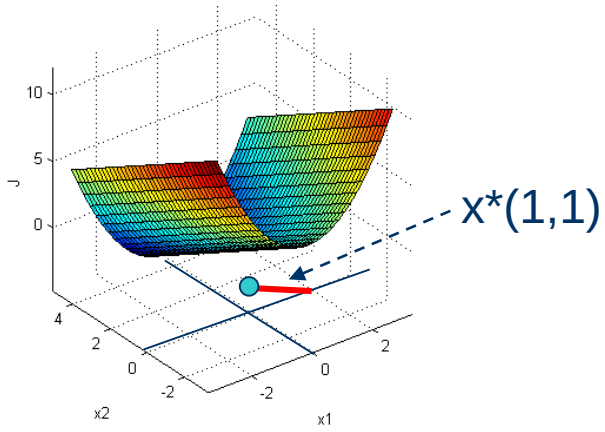
First order sufficient KKT conditions

$$\min_x J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \leq \mathbf{0}$$

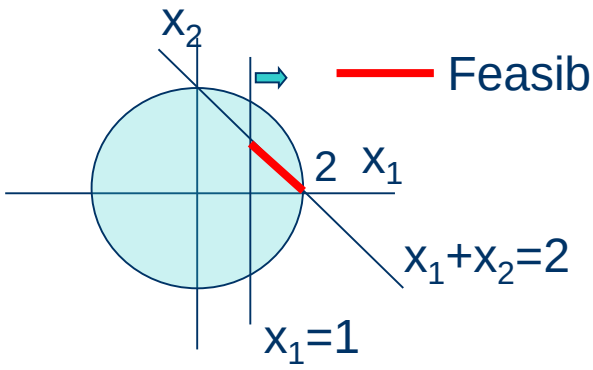
If the function J is convex, the constraints g_i are convex and the equality constraints h_j are linear, then the feasible set is convex and there exist a solution of the KKT conditions that corresponds to the global optimum of the NLP problem

$$\nabla_x J(\mathbf{x}) + \sum_j \lambda_j \nabla_x h_j(\mathbf{x}) + \sum_i \mu_i \nabla_x g_i(\mathbf{x}) = 0$$
$$h_j(\mathbf{x}) = 0$$
$$g_i(\mathbf{x}) \leq 0$$
$$\mu_i g_i(\mathbf{x}) = 0$$
$$\mu_i \geq 0$$

Example 2



$$\left.\begin{aligned} &\min_{x_1,x_2} x_1^2 - x_2 \\ &x_1 + x_2 = 2 \\ &x_1^2 + x_2^2 \le 4 \\ &-x_1 + 1 \le 0 \end{aligned}\right\}$$



Does it fulfil the first order sufficient conditions? Yes

$$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = x_1^2 - x_2 + \lambda(x_1 + x_2 - 2) + \mu_1(x_1^2 + x_2^2 - 4) + \mu_2(-x_1 + 1)$$

$$\nabla_x J(\mathbf{x}) + \sum_j \lambda_j \nabla_x h_j(\mathbf{x}) + \sum_i \mu_i \nabla_x g_i(\mathbf{x}) = 0 \qquad 2x_1 + \lambda + \mu_1 2x_1 - \mu_2 = 0$$

$$-1 + \lambda + \mu_1 2x_2 = 0$$

$$h_j(\mathbf{x}) = 0 \qquad \Rightarrow \qquad x_1 + x_2 = 2 \quad \mu_1 \ge 0 \quad \mu_2 \ge 0$$

$$g_i(\mathbf{x}) \le 0 \qquad x_1^2 + x_2^2 \le 4 \quad \mu_1(x_1^2 + x_2^2 - 4) = 0$$

$$\mu_i g_i(\mathbf{x}) = 0 \qquad -x_1 + 1 \le 0 \quad \mu_2(-x_1 + 1) = 0$$

$$\mu_i \ge 0$$

Example 2

$$
\left.
\begin{array}{l}
2x_1 + \lambda + \mu_1 2x_1 - \mu_2 = 0 \\
-1 + \lambda + \mu_1 2x_2 = 0 \\
x_1 + x_2 = 2 \quad \mu_1 \geq 0 \quad \mu_2 \geq 0 \\
x_1^2 + x_2^2 \leq 4 \quad \mu_1 (x_1^2 + x_2^2 - 4) = 0 \\
-x_1 + 1 \leq 0 \quad \mu_2 (-x_1 + 1) = 0
\end{array}
\right\}
$$

$$x_1 = \frac{\mu_2 - \lambda}{2(1+\mu_1)}$$

$$x_2 = \frac{1-\lambda}{2\mu_1}$$

$$\frac{\mu_2 - \lambda}{2(1+\mu_1)} + \frac{1-\lambda}{2\mu_1} = 2$$

$$\left[\left(\frac{\mu_2 - \lambda}{2(1+\mu_1)} \right)^2 + \left(\frac{1-\lambda}{2\mu_1} \right)^2 - 4 \right] \mu_1 = 0$$

$$\left[\frac{-\mu_2 + \lambda}{2(1+\mu_1)} + 1 \right] \mu_2 = 0$$

All alternatives must be examined in order to find the solution

$$x_1 = \frac{\mu_2 - \lambda}{2(1+\mu_1)}$$

$$x_2 = \frac{1-\lambda}{2\mu_1}$$

$$\frac{\mu_2 - \lambda}{2(1+\mu_1)} + \frac{1-\lambda}{2\mu_1} = 2$$

$$\left[\left(\frac{\mu_2 - \lambda}{2(1+\mu_1)}\right)^2 + \left(\frac{1-\lambda}{2\mu_1}\right)^2 - 4\right]\mu_1 = 0$$

$$\left[\frac{-\mu_2 + \lambda}{2(1+\mu_1)} + 1\right]\mu_2 = 0$$

if $\mu_1 = 0, \mu_2 = 0$

$$\begin{cases} \dfrac{\mu_2 - \lambda}{2(1+\mu_1)} + \dfrac{1-\lambda}{2\mu_1} = 2 \end{cases}$$

if $\mu_1 = 0, \mu_2 \neq 0$

$$\begin{cases} \dfrac{\mu_2 - \lambda}{2(1+\mu_1)} + \dfrac{1-\lambda}{2\mu_1} = 2 \\ \dfrac{-\mu_2 + \lambda}{2(1+\mu_1)} + 1 = 0 \end{cases}$$

if $\mu_2 = 0, \mu_1 \neq 0 \Rightarrow$

$$\begin{cases} \left(\dfrac{-\lambda}{2(1+\mu_1)}\right)^2 + \left(\dfrac{1-\lambda}{2\mu_1}\right)^2 - 4 = 0 \\ \dfrac{-\lambda}{2(1+\mu_1)} + \dfrac{1-\lambda}{2\mu_1} = 2 \end{cases}$$

if $\mu_1 \neq 0, \mu_2 \neq 0$

$$\begin{cases} \dfrac{\mu_2 - \lambda}{2(1+\mu_1)} + \dfrac{1-\lambda}{2\mu_1} = 2 \\ \left(\dfrac{\mu_2 - \lambda}{2(1+\mu_1)}\right)^2 + \left(\dfrac{1-\lambda}{2\mu_1}\right)^2 - 4 = 0 \\ \dfrac{-\mu_2 + \lambda}{2(1+\mu_1)} + 1 = 0 \end{cases}$$

How to solve the case $\mu_2=0$, $\mu_1\neq 0$

$$\left.\begin{array}{l}\left(\dfrac{-\lambda}{2(1+\mu_1)}\right)^2+\left(\dfrac{1-\lambda}{2\mu_1}\right)^2-4=0\\ \dfrac{-\lambda}{2(1+\mu_1)}+\dfrac{1-\lambda}{2\mu_1}=2\end{array}\right\}\quad \dfrac{-\lambda}{2(1+\mu_1)}=2-\dfrac{1-\lambda}{2\mu_1}$$

$$\left(2-\frac{1-\lambda}{2\mu_1}\right)^2+\left(\frac{1-\lambda}{2\mu_1}\right)^2=4$$

$$4+\left(\frac{1-\lambda}{2\mu_1}\right)^2-4\frac{1-\lambda}{2\mu_1}+\left(\frac{1-\lambda}{2\mu_1}\right)^2=4 \qquad ((1-\lambda)-4\mu_1)(1-\lambda)=0 \begin{cases}\lambda=1\\ 1-\lambda=4\mu_1\end{cases}$$

$$\lambda=1\rightarrow\frac{-1}{2(1+\mu_1)}=2 \quad 1+\mu_1=-1/4 \quad \mu_1=-5/4 \; NF$$

$$1-\lambda=4\mu_1\rightarrow\frac{4\mu_1-1}{2(1+\mu_1)}+\frac{4\mu_1}{2\mu_1}=2 \quad \frac{4\mu_1-1}{2(1+\mu_1)}+2=2\rightarrow\mu_1=1/4,\lambda=0$$

$$x_1=0, x_2=2 \quad NF$$

Example 2

$$x_1 = \frac{\mu_2 - \lambda}{2(1+\mu_1)}$$

$$x_2 = \frac{1-\lambda}{2\mu_1}$$

$$\frac{\mu_2 - \lambda}{2(1+\mu_1)} + \frac{1-\lambda}{2\mu_1} = 2$$

$$\left[\left(\frac{\mu_2 - \lambda}{2(1+\mu_1)}\right)^2 + \left(\frac{1-\lambda}{2\mu_1}\right)^2 - 4\right]\mu_1 = 0$$

$$\left[\frac{-\mu_2 + \lambda}{2(1+\mu_1)} + 1\right]\mu_2 = 0$$

$$\begin{cases}
\text{if } \mu_1 = 0, \mu_2 = 0 & \{\lambda = 1, x_1 = -1/2, x_2 = 5/2 \text{ NF} \\
\text{if } \mu_1 = 0, \mu_2 \neq 0 & \{\lambda = 1, \mu_2 = 3, x_1 = 1, x_2 = 1 \\
\text{if } \mu_1 \neq 0, \mu_2 = 0 & \begin{cases} \mu_1 = 1/4, \lambda = 0, x_1 = 0, x_2 = 2 \text{ NF} \\ \mu_1 = -5/4, \lambda = 1 \text{ NF} \end{cases} \\
\text{if } \mu_1 \neq 0, \mu_2 \neq 0 & \{\lambda = 1; \mu_1 = 0, \mu_2 = 3, x_1 = 1, x_2 = 1
\end{cases}$$

Optimal solution

Constraint qualifications

$$\min_{x_1,x_2} x_1^2 - x_2$$
$$x_1 + x_2 = 2$$
$$x_1^2 + x_2^2 \le 4$$
$$-x_1 + 1 \le 0$$

Active constraints

Optimum: $x_1=1, x_2=1, \lambda=1, \mu_1=0, \mu_2=3$ J = 0

$\nabla_x h(x^*)$, $\nabla_x g(x^*)$ (active)

$$\nabla_x h = \begin{bmatrix} 1 \\ 1 \end{bmatrix}' \quad \nabla_x g = \begin{bmatrix} -1 \\ 0 \end{bmatrix}' \quad \text{rank} \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} = 2$$

There are linearly independent, so the constraint qualification is fulfilled

Sensitivity

$$\min_x \; J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{b}$$
$$\mathbf{g(x)} \leq \mathbf{c}$$

From the interpretation of the KKT as Lagrange multipliers, it is possible to formulate directly the following relations:

$$\frac{\partial J(\mathbf{x}^*)}{\partial \mathbf{b}} = -\boldsymbol{\lambda}^*, \quad \frac{\partial J(\mathbf{x}^*)}{\partial \mathbf{c}} = -\boldsymbol{\mu}^*,$$

These relations allow us to compute how the optimal cost changes when the constraints are relaxed by an unit, which is a very important information in decision making problems

Example 2

$$\min_{x_1,x_2} J(x_1,x_2) = \min_{x_1,x_2} x_1^2 - x_2$$

$$x_1 + x_2 = 2$$

$$x_1^2 + x_2^2 \le 4$$

$$-x_1 + 1 \le 0$$

Active constraints

Optimum: $x_1=1, x_2=1, \lambda=1, \mu_1=0, \mu_2=3$, $J=0$

How much changes the optimal J if the right hand side of the constraints is changed by 0.1 units?

Sensitivities: -1,0,-3

$$\frac{\partial J}{\partial b} = -\lambda$$

$$\Delta J \approx \frac{\partial J}{\partial b}\Delta b$$

If each one would be increased by an unit, keeping the other constant, then the first one would decrease 1*0.1 units the cost function, the second one would not change it and the third one would decrease the optimal cost by 3*0.1 units (in a linear approximation).

Second order sufficient KKT conditions (SOSC)

In order to assure that the solution given by the first order KKT conditions is a minimum, and not a maximum or a saddle point of the Lagrangian $L(x,\lambda,\mu)$ in (x^*,λ^*,μ^*), the second order conditions are formulated.

Second order conditions provide **sufficient** optimality conditions for the solution and are given by the following expression involving the Hessian $\nabla^2_x L$:

$$\mathbf{z}'\nabla_x^2 L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)\mathbf{z} > 0 \qquad \forall \mathbf{z} \neq 0 \text{ such that } \begin{cases} \nabla_x \mathbf{h}(\mathbf{x}^*)\mathbf{z} = 0 \\ \nabla_x g_{act,i}(\mathbf{x}^*)\mathbf{z} = 0 \quad \mu_i > 0 \\ \nabla_x g_{act,i}(\mathbf{x}^*)\mathbf{z} \leq 0 \quad \mu_i = 0 \end{cases}$$

So, the Hessian of L with respect to x is required to be PD only in the direction of all vectors z orthogonal to the gradients of the active constraints at x*. The necessary condition only required the Hessian to be PSD. (g_{act} stands for the active constraints)

Lagrangian $L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = x_1^2 - x_2 + \lambda(x_1 + x_2 - 2) + \mu_1(x_1^2 + x_2^2 - 4) + \mu_2(-x_1 + 1)$

$$\nabla_x J(\mathbf{x}) + \sum_j \lambda_j \nabla_x h_j(\mathbf{x}) + \sum_i \mu_i \nabla_x g_i(\mathbf{x}) = 0 \quad \begin{cases} 2x_1 + \lambda + \mu_1 2x_1 - \mu_2 = 0 \\ -1 + \lambda + \mu_1 2x_2 = 0 \end{cases}$$

Example 2

Optimum: $x_1=1, x_2=1, \lambda=1, \mu_1=0, \mu_2=3$

$$\begin{aligned} &\min_{x_1,x_2} x_1^2 - x_2 \\ &x_1 + x_2 = 2 \\ &x_1^2 + x_2^2 \le 4 \\ &-x_1 + 1 \le 0 \end{aligned}$$

Active constraints

$$\mathbf{z}' \nabla_x^2 L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\mathbf{z} > 0 \qquad \forall z \neq 0 \mid \begin{bmatrix} \nabla_x \mathbf{h}(\mathbf{x}^*) \\ \nabla_x \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix} \mathbf{z} = \mathbf{0}$$

$$\begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 0 \quad z_1 + z_2 = 0$$

$$-z_1 = 0$$

$$\mathbf{z} = \mathbf{0}$$

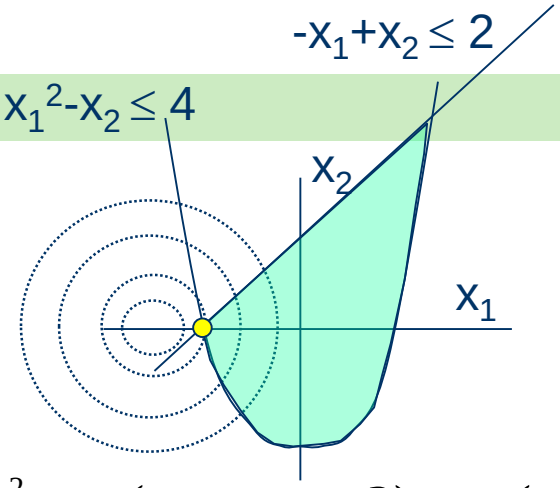
Hessian with respect to x

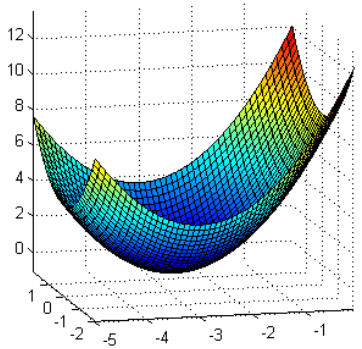
$$H_x = \begin{bmatrix} \frac{\partial L^2}{\partial x_1^2} & \frac{\partial L^2}{\partial x_1 \partial x_2} \\ \frac{\partial L^2}{\partial x_2 \partial x_1} & \frac{\partial L^2}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 + 2\mu_1 & 0 \\ 0 & 2\mu_1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

There are no vectors of tangent planes perpendicular to all active constraints different from zero to test the SOSC. So, it is satisfied in a vacuous way.

Example 3

$$\left.\begin{aligned}\min_{x_1,x_2}\ & (x_1+3)^2+x_2^2\\ & -x_1+x_2-2\le 0\\ & x_1^2-x_2-4\le 0\end{aligned}\right\}$$





$$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})=(x_1+3)^2+x_2^2+\mu_1(-x_1+x_2-2)+\mu_2(x_1^2-x_2-4)$$

$$\nabla_x J(\mathbf{x})+\sum_j \lambda_j \nabla_x h_j(\mathbf{x})+\sum_i \mu_i \nabla_x g_i(\mathbf{x})=0$$

$$h_j(\mathbf{x})=0$$

$$g_i(\mathbf{x})\le 0$$

$$\mu_i g_i(\mathbf{x})=0$$

$$\mu_i \ge 0$$

$$2(x_1+3)+\mu_1(-1)+\mu_2 2x_1=0$$

$$2x_2+\mu_1(1)+\mu_2(-1)=0$$

$$(-x_1+x_2-2)\mu_1=0 \quad \mu_1\ge 0$$

$$(x_1^2-x_2-4)\mu_2=0 \quad \mu_2\ge 0$$

$$-x_1+x_2-2\le 0$$

$$x_1^2-x_2-4\le 0$$

Example 3

$$
\left.\begin{aligned}
&2(x_1+3)+\mu_1(-1)+\mu_2 2x_1=0\\
&2x_2+\mu_1(1)+\mu_2(-1)=0\\
&(-x_1+x_2-2)\mu_1=0 \quad \mu_1\geq 0\\
&(x_1^2-x_2-4)\mu_2=0 \quad \mu_2\geq 0\\
&-x_1+x_2-2\leq 0\\
&x_1^2-x_2-4\leq 0
\end{aligned}\right\}
\qquad
\left.\begin{aligned}
x_1&=\frac{\mu_1-6}{2(1+\mu_2)}\\
x_2&=\frac{\mu_2-\mu_1}{2}\\
&\left(\frac{6-\mu_1}{2(1+\mu_2)}+\frac{\mu_2-\mu_1}{2}-2\right)\mu_1=0\\
&\left(\left(\frac{\mu_1-6}{2(1+\mu_2)}\right)^2-\frac{\mu_2-\mu_1}{2}-4\right)\mu_2=0
\end{aligned}\right\}
$$

if $\mu_1=0,\mu_2=0 \quad \rightarrow x_1=-3,x_2=0 \quad$ NF

if $\mu_1\neq 0,\mu_2=0 \quad \left\{\frac{6-\mu_1}{2}+\frac{-\mu_1}{2}-2=0 \quad \rightarrow \mu_1=1,x_1=-5/2,x_2=-1/2\right.$ NF

Example 3

$$\left.\begin{array}{l} x_1 = \dfrac{\mu_1 - 6}{2(1+\mu_2)} \\ x_2 = \dfrac{\mu_2 - \mu_1}{2} \\ \left(\dfrac{6-\mu_1}{2(1+\mu_2)} + \dfrac{\mu_2-\mu_1}{2} - 2\right)\mu_1 = 0 \\ \left(\left(\dfrac{\mu_1-6}{2(1+\mu_2)}\right)^2 - \dfrac{\mu_2-\mu_1}{2} - 4\right)\mu_2 = 0 \end{array}\right\} \text{if } \mu_1 = 0, \mu_2 \neq 0 \quad \begin{cases} \dfrac{36}{4(1+\mu_2)^2} - \dfrac{\mu_2}{2} = 4 \\ \mu_2^3 + 10\mu_2^2 + 17\mu_2 = 10 \\ \mu_2 = -7.58, -2.87, 0.46 \\ x_1 = -2.06,\ x_2 = 0.23 \text{ NF} \end{cases}$$

$$\text{if } \mu_1 \neq 0, \mu_2 \neq 0 \begin{cases} \dfrac{6-\mu_1}{2(1+\mu_2)} + \dfrac{\mu_2-\mu_1}{2} - 2 = 0 \\ \left(\dfrac{\mu_1-6}{2(1+\mu_2)}\right)^2 - \dfrac{\mu_2-\mu_1}{2} - 4 = 0 \end{cases}$$

$$\left(\frac{\mu_2-\mu_1}{2} - 2\right)^2 - \frac{\mu_2-\mu_1}{2} = 4 \rightarrow (\mu_2-\mu_1)^2 = 10(\mu_2-\mu_1)$$

$$\begin{cases} \mu_2 = \mu_1 \rightarrow \mu_1 = 2/5,\ \mu_2 = 2/5, & x_1 = -2,\ x_2 = 0 \\ \mu_2 - \mu_1 = 10 \rightarrow \dfrac{6-\mu_1}{11+\mu_1} + 6 = 0 \rightarrow \mu_1 = -14.4\ NF \end{cases}$$

Optimum as the region and J are convex

Constraint qualifications

$$\min_{x_1,x_2} (x_1+3)^2 + x_2^2$$
$$-x_1 + x_2 - 2 \le 0$$
$$x_1^2 - x_2 - 4 \le 0$$

Active constraints

Optimum: x_1=-2,x_2=0,μ_1=2/5,μ_2=2/5, J=1

$\nabla_x h(x^*)$, $\nabla_x g(x^*)$ (active)

$$\nabla_x g_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \nabla_x g_2 = \begin{bmatrix} 2(-2) \\ -1 \end{bmatrix} \quad \text{rank} \begin{bmatrix} -1 & -4 \\ 1 & -1 \end{bmatrix} = 2$$

There are linearly independent, so the constraint qualification is fulfilled

Sensitivities

$$\min_{x_1, x_2} (x_1 + 3)^2 + x_2^2$$

$$-x_1 + x_2 - 2 \leq 0$$

$$x_1^2 - x_2 - 4 \leq 0$$

Active constraints

Optimum: x_1=-2,x_2=0,μ_1=2/5,μ_2=2/5, J=1

How much changes the optimal J if the right hand side of the constraints is changed by an unit?

Sensitivities: -2/5,-2/5

If each right hand side were increased by one unit, maintaining the other constant, J would improve (decrease) by 2/5 in each case

$$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = (x_1+3)^2 + x_2^2 + \mu_1(-x_1+x_2-2) + \mu_2(x_1^2 - x_2 - 4)$$

$$\nabla_x J(\mathbf{x}) + \sum_j \lambda_j \nabla_x h_j(\mathbf{x}) + \sum_i \mu_i \nabla_x g_i(\mathbf{x}) = 0 \quad \begin{cases} 2(x_1+3) + \mu_1(-1) + \mu_2 2x_1 = 0 \\ 2x_2 + \mu_1(1) + \mu_2(-1) = 0 \end{cases}$$

Ejemplo 3

Optimum: $x_1=-2, x_2=0, \mu_1=2/5, \mu_2=2/5$, J=1

2º order Conditions

$$\min_{x_1,x_2} (x_1+3)^2 + x_2^2$$

$$-x_1 + x_2 - 2 \le 0 \quad \leftarrow \text{Active constraints}$$

$$x_1^2 - x_2 - 4 \le 0 \quad \leftarrow \text{Active constraints}$$

$$\mathbf{z}'\nabla_x^2 L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)\mathbf{z} > 0 \qquad \forall z \mid \begin{bmatrix} \nabla_x \mathbf{h}(\mathbf{x}^*) \\ \nabla_x \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix} \mathbf{z} = \mathbf{0}$$

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 0 \quad -z_1 + z_2 = 0$$

$$\begin{bmatrix} 2(-2) & -1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 0 \quad -4z_1 - z_2 = 0$$

Vectors of tangent planes

z = 0

Hessiano respecto a x

$$H_x = \begin{bmatrix} \frac{\partial L^2}{\partial x_1^2} & \frac{\partial L^2}{\partial x_1 \partial x_2} \\ \frac{\partial L^2}{\partial x_2 \partial x_1} & \frac{\partial L^2}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2+2\mu_2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 14/5 & 0 \\ 0 & 2 \end{bmatrix}$$

There are no vectors of tangent planes to the active constraints different from zero to test the SOOC: But as H_x is PD, any quadratic form is convex in all directions and the SOSC are satisfied in all directions.

Example 4

$$\left.\begin{array}{l} \min\limits_{x_1,x_2} x_2^2 - 0.1(x_1-4)^2 \\ 1-\left(x_1^2+x_2^2\right)\le 0 \end{array}\right\}$$

$$L = x_2^2 - 0.1(x_1-4)^2 + \mu\left(1-x_1^2-x_2^2\right)$$

$$\nabla L_x = \left[-0.2(x_1-4)-2\mu x_1 \quad (2-2\mu)x_2\right]$$

$$-0.2(x_1-4)-2\mu x_1 = 0$$

$$(2-2\mu)x_2 = 0$$

$$\mu\left(1-x_1^2-x_2^2\right)=0$$

2 possible KKT solutions:

$x_1 = 1 \quad x_2 = 0 \quad \mu = 0.3$

$x_1 = 4 \quad x_2 = 0 \quad \mu = 0$

$$\nabla L_{xx} = \begin{bmatrix} -0.2-2\mu & 0 \\ 0 & 2-2\mu \end{bmatrix}$$

SOSC are necessary to delucidade which solutions are correct

Example 4

$$\left.\begin{array}{l} \min\limits_{x_1,x_2} x_2^2 - 0.1(x_1-4)^2 \\ 1-\left(x_1^2+x_2^2\right)\le 0 \end{array}\right\}$$

$x_1 = 4 \quad x_2 = 0 \quad \mu = 0 \qquad J = 0$

Inactive constraint

There is no g to verify the SOSC

$$\mathbf{z}'\nabla_x^2 L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)\mathbf{z} > 0 \qquad \forall z \neq 0 \mid \begin{bmatrix} \nabla_x \mathbf{h}(\mathbf{x}^*) \\ \nabla_x \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix}\mathbf{z} = \mathbf{0}$$

$$\nabla L_{xx} = \begin{bmatrix} -0.2-2\mu & 0 \\ 0 & 2-2\mu \end{bmatrix}$$

$$\nabla L_{xx}\left(\mathbf{x}^*,\mu^*\right) = \begin{bmatrix} -0.2 & 0 \\ 0 & 2 \end{bmatrix}$$

Nevertheless, as zHz is ND for all directions z, this solution corresponds to a local maximum

Example 4

SOSC

$$\mathbf{z}'\nabla_x^2 L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)\mathbf{z} > 0 \qquad \forall z \mid \begin{bmatrix} \nabla_x \mathbf{h}(\mathbf{x}^*) \\ \nabla_x \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix}\mathbf{z} = \mathbf{0}$$

$$\min_{x_1,x_2} x_2^2 - 0.1(x_1-4)^2$$
$$1-(x_1^2+x_2^2)\le 0$$

$x_1 = 1 \quad x_2 = 0 \quad \mu = 0.3 \quad J = -0.9$

Active constraint

$$g(\mathbf{x}) = (1-x_1^2-x_2^2)$$

$$\nabla g(\mathbf{x}^*)\mathbf{z} = 0 \quad \Rightarrow \begin{bmatrix} -2x_1^* & -2x_2^* \end{bmatrix}\mathbf{z} = \begin{bmatrix} -2 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 0 \Rightarrow z_1 = 0 \quad \mathbf{z} = \begin{bmatrix} 0 \\ z \end{bmatrix}$$

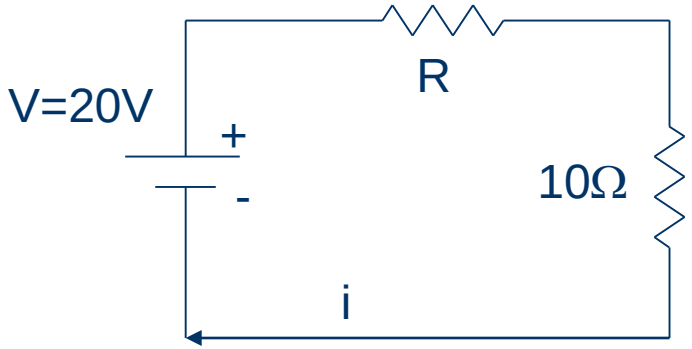
$$\nabla L_{xx} = \begin{bmatrix} -0.2-2\mu & 0 \\ 0 & 2-2\mu \end{bmatrix}$$

$$\mathbf{z}'\nabla L_{xx}\mathbf{z} = \begin{bmatrix} 0 & z \end{bmatrix}\begin{bmatrix} -0.8 & 0 \\ 0 & 1.4 \end{bmatrix}\begin{bmatrix} 0 \\ z \end{bmatrix} =$$

$$\nabla L_{xx}(\mathbf{x}^*,\mu^*) = \begin{bmatrix} -0.8 & 0 \\ 0 & 1.4 \end{bmatrix}$$

$$= 1.4z^2 > 0$$ SOSC are verified

Example 5



Find the value of R so that the power dissipated in this resistor is maximized

$$\left.\begin{aligned} P &= I^2 R \\ 20 &= IR + 10I \end{aligned}\right\} \min_{R} \left\{ \frac{400R}{(R+10)^2} \right\} \quad R \geq 0$$

Solving NLP problems

- The analytical solution of a NLP problem is only possible in very simple problems, but it provides the theoretical framework.
- There are several practical alternatives for the numerical solution of NLP problems:
  - Use the duality theory
  - Active set methods
  - Exploit its particular mathematical structure (QP, Wolfe,..)
  - Transform the problem in another one without constraints that has the same solution (Penalty functions)
  - Solve a series of approximate simpler problems until the solution is found (SLP, Cutting Plane,..)
  - Solve the KKT conditions by a succession of approximate problems (SQP)
  - Simplify the problem eliminating variables by means of the equality constraints (GRG)
  - Interior point methods,
  - Etc.

Active set Methods

The idea behind active set methods is to partition inequality constraints into two sets: those that are considered active and those considered inactive. The active constraints can be treated as equality ones, while the inactive can be, consequently, ignored. The key point is to decide how to incorporate or exclude a constraint to or from the active set.

For simplicity, only inequality constraints will be considered. Equality ones can be incorporated easily.

$$\left.\begin{array}{l}\min\limits_{\mathbf{x}} J(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\end{array}\right\}$$

Be A the subset of indexes of the active constraints. Then the KKT conditions are:

$$\nabla_x J(\mathbf{x}) + \sum_{i \in A} \mu_i \nabla_x g_i(\mathbf{x}) = 0$$

$$g_i(\mathbf{x}) = 0 \quad i \in A$$

$$\mu_i \geq 0 \quad i \in A$$

$$g_i(\mathbf{x}) < 0 \quad i \notin A$$

$$\mu_i = 0 \quad i \notin A$$

Active set Methods

If the active set were known, the problem could be solved as an equality constrained one. As the active set is not known before solving the problem, we can guess one, called the working set, W, and solve the corresponding equality constrained problem.

The working set W is chosen as a subset of the active constraints for the current point x_k in order to guarantee feasibility.

If the corresponding Lagrange multipliers μ of the solution are non-negative, and the other constraints are satisfied, then the solution found is the optimal.

If not, for every $i \in W$ such that $\mu_i < 0$, the value of J can be decreased by relaxing this constraint, removing it from the working set.

During the intermediate iterations of the algorithm, as x_k changes, it is necessary to check that the remaining constraints are satisfied. If a new constraint becomes active, it should be incorporated to the working set

Quadratic Programming (QP)

A special set of optimization problems with many practical applications corresponds to the case when the cost function is quadratic in the decision variables, but the constraints are linear. This type of problems is known as Quadratic Programming (QP).

$$\min_x J(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

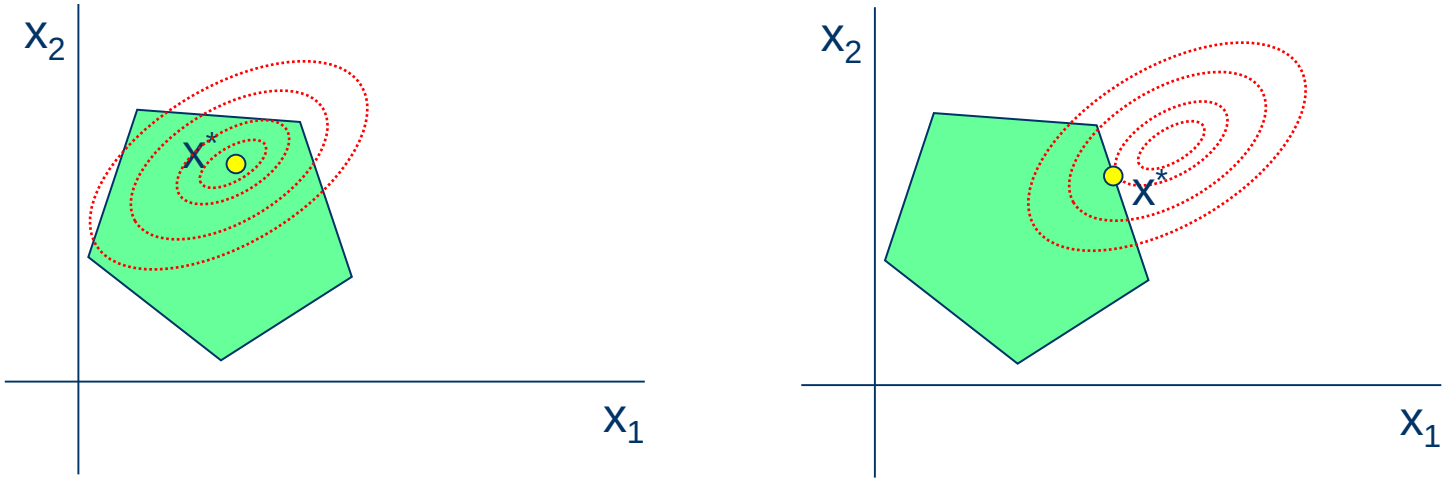
$$\mathbf{x} \geq 0$$

Q (n x n) symmetric

A (m x n) rank m < n

Other equivalent formulations can be found where the constraints are expressed as inequalities, or the cost as maximization, etc. The conversion between the different types can be made using the same techniques as in LP.

Quadratic Programming (QP)



Unlike LP, in the case of QP the optimum x* can be located either in the border or in the interior of the feasible set

Quadratic Programming (QP)

$$\min_{x} J(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x}$$

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$$

$$-\mathbf{x} \leq 0$$

Several solutions methods:

- ✓Solving the KKT conditions
- ✓Active sets
- ✓More general methods (Wolfe)
- ✓.....

Being A of rank m, the gradients $\nabla_x h = A$, $\nabla_x g = -I$ are independent and the constraint qualifications are always satisfied.

As the feasible set, if non empty, is convex, it happens that if the matrix Q is positive semi-definite, then J is convex in a convex set and any local solution is a global one, which can be found solving the KKT conditions.

If Q is not PSD, then several or none local optimums can exist and the KKT conditions are only necessary ones.

Solving QP using the KKT conditions
(Dantzing-Wolfe)

$$\min_x J(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x}$$

$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$$

$$-\mathbf{x} \leq 0$$

Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x} + \boldsymbol{\lambda}'(\mathbf{A}\mathbf{x} - \mathbf{b}) - \boldsymbol{\mu}'\mathbf{x}$$

KKT Conditions :

$$\mathbf{c} + \mathbf{Q}\mathbf{x} + \mathbf{A}'\boldsymbol{\lambda} - \boldsymbol{\mu} = \mathbf{0}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \boldsymbol{\mu} \geq \mathbf{0}$$

$$\boldsymbol{\mu}'\mathbf{x} = 0$$

It is possible to find a feasible solution of these linear set of equations in (x,λ,μ) using a simplex phase I approach while forcing the condition **μ'x=0** in practice by imposing that columns corresponding to components of the vector (x,λ,μ) different from zero in both x and μ are not in the basis simultaneously, during the pivoting.

Linear equations excepting **μ'x = 0,** ⇒ x_i or μ_i are zero.

Solving the KKT conditions

$$\left.\begin{array}{l} \mathbf{c}+\mathbf{Q}\mathbf{x}+\mathbf{A}'\boldsymbol{\lambda}-\boldsymbol{\mu}=\mathbf{0} \\ \mathbf{A}\mathbf{x}=\mathbf{b} \\ \mathbf{x}\geq\mathbf{0}, \quad \boldsymbol{\mu}\geq\mathbf{0} \\ \boldsymbol{\mu}'\mathbf{x}=0 \end{array}\right\} \Rightarrow \begin{array}{c} \boldsymbol{\lambda}=\boldsymbol{\sigma}-\boldsymbol{\delta} \\ \boldsymbol{\sigma}\geq\mathbf{0}, \boldsymbol{\delta}\geq\mathbf{0} \\ \mathbf{z}'=(\mathbf{x},\boldsymbol{\sigma},\boldsymbol{\delta},\boldsymbol{\mu}) \end{array}$$

The condition $\mu'x=0$ is imposed by means of rules that select the appropriate columns in the pivoting operations

$$\left.\begin{array}{l} \mathbf{c}+\mathbf{Q}\mathbf{x}+\mathbf{A}'(\boldsymbol{\sigma}-\boldsymbol{\delta})-\boldsymbol{\mu}=\mathbf{0} \\ \mathbf{A}\mathbf{x}=\mathbf{b} \\ \mathbf{z}\geq\mathbf{0} \end{array}\right\} \begin{bmatrix} \mathbf{Q} & \mathbf{A}' & -\mathbf{A}' & -\mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\sigma} \\ \boldsymbol{\delta} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b} \end{bmatrix}, \quad \mathbf{z}\geq\mathbf{0}$$

Mz = β
z ≥ 0

$$\left.\begin{array}{l} \max\limits_{\mathbf{z},\boldsymbol{\nu}} -(1,1,\ldots 1)\boldsymbol{\nu} \\ \begin{bmatrix} \mathbf{I} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu} \\ \mathbf{z} \end{bmatrix} = \boldsymbol{\beta}, \quad \mathbf{z}\geq\mathbf{0}, \boldsymbol{\nu}\geq 0 \end{array}\right\}$$

LP phase I problem, which solution if $\nu^*=0$, provides the solution to the KKT conditions

Active sets

This method is oriented to solving problems where the constraints are linear. The idea behind is the fact that, in every step of the algorithm, the active set can be considered as a set of equality constraints, while the non-active set can be ignored.

$$\min_{\mathbf{x}} J(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$

Then, at every step of the algorithm, the problem to solve is:

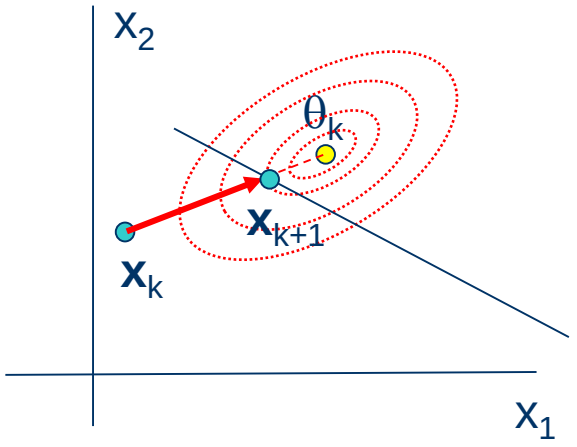
$$\left.\begin{aligned} &\min_{\mathbf{x}} J(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x} \\ &\mathbf{a_i}'\mathbf{x} - \beta_i = 0 \quad i \in \Lambda \end{aligned}\right\} P_\Lambda$$

As there are no inequalities, the problem can be solved easily by substitution or Lagrange multipliers

Where Λ is the set of indexes of the active set

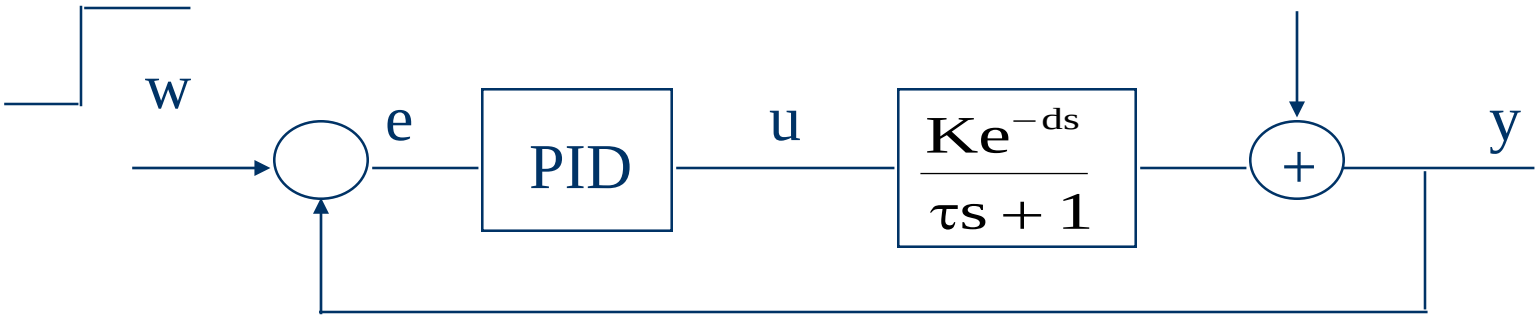
Active sets

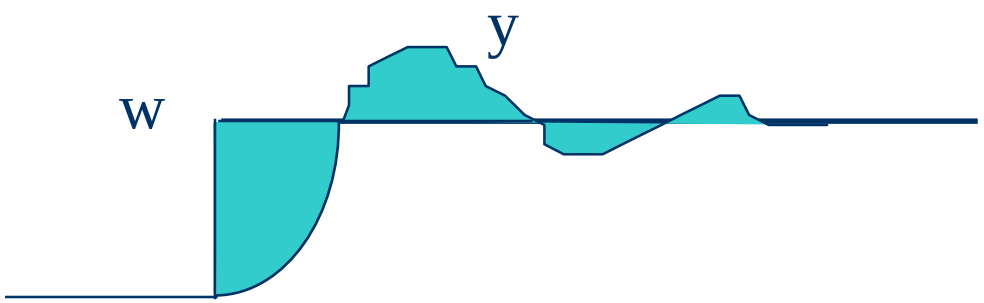
1. Choose x_k and select Λ
2. Solve the problem, P_Λ, associated to equalities in Λ. Let θ_k be the solution and λ_k its Lagrange multipliers.
3. If the λ_k are all ≥ 0, and θ_k satisfy all constraints, then θ_k is optimum.
4. Conversely, remove the indexes p, corresponding to those $\lambda_k < 0$ from Λ
5. If θ_k does not verify a constraint that is not in Λ, compute $x_{k+1}=x_k+\alpha_k(\theta_k-x_k)$ with ($\alpha_k <1$) and add the corresponding index to Λ
6. Do k=k+1 and return to 2



$$\alpha_k = \min\left(1, \min_{\substack{i\notin\Lambda \\ \mathbf{a_i}'(\boldsymbol{\theta_k}-\mathbf{x}_k)<0}} \frac{\beta_i - \mathbf{a_i}'\mathbf{x_k}}{\mathbf{a_i}'(\boldsymbol{\theta}_k - \mathbf{x}_k)}\right)$$

Example QP (Dynamic) Optimal controller tuning





$$\min_{K_p,T_i,T_d} \int e(t)^2 dt \quad \text{MISE}$$

$$K_p \geq 0, T_i \geq 0, T_d \geq 0$$

error = w - y (linear function of K_p , T_i, T_d)

Successive linear programming SLP

A natural extension of the Wolfe method that can be applied to a general NLP problem consist in linearizing J(x) and the constraints around a point x_k forming a LP problem which solution x_k^* should be closer to the optimal NLP than the starting point. Next, one makes $x_{k+1}=x_k^*$ and the procedure is repeated until an ending condition is met

$$\left.\begin{array}{l}\min\limits_x J(\mathbf{x}) \\ \mathbf{h(x)=0} \\ \mathbf{g(x)\le 0}\end{array}\right\} \approx \left.\begin{array}{l}\min\limits_x J(\mathbf{x}_k)+\nabla_x J(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) \\ \mathbf{h}(\mathbf{x}_k)+\nabla_x \mathbf{h}(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)=\mathbf{0} \\ \mathbf{g}(\mathbf{x}_k)+\nabla_x \mathbf{g}(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)\le \mathbf{0}\end{array}\right\} \mathbf{m}\le \mathbf{x}-\mathbf{x}_k \le \mathbf{M}$$

Some extra constraint on the maximum change around the linearization point are included in order to guarantee an acceptable approximation

Penalty methods

The core idea is to transform the NLP problem

$$\min_x J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \leq \mathbf{0}$$

Into another one without constraints, which solution is approximately equal to the one of the original NLP

$$\min_x V(\mathbf{x}, \eta) = \min_x J(\mathbf{x}) + \sum_i \eta_i P_i(h_i(\mathbf{x}), g_i(\mathbf{x}))$$

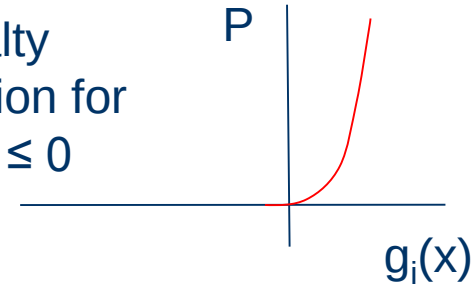
Here V is the new unconstraint function to be minimized, which is formed adding the penalty functions P_i to the original cost function J. η_i are adjustable parameters whose values are changed along the iterations in order to approach the minimum of V to the NLP optimum

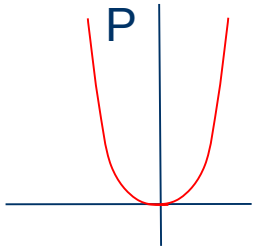
Penalty functions

$$\min_x V(\mathbf{x}, \eta) = \min_x J(\mathbf{x}) + \sum_i \eta_i P_i(h_i(\mathbf{x}), g_i(\mathbf{x}))$$

The main characteristic of the penalty functions P is that they tend to be zero when a constraint is satisfied, but take a large number when the constraint is violated or closed to be violated. In this way the algorithm minimizing V will not search in the regions where the constraints are violated. The penalties P change the shape of the original cost function J, increasing its value in those regions where the constraints are not satisfied.

Penalty function for $g_i(x) \leq 0$



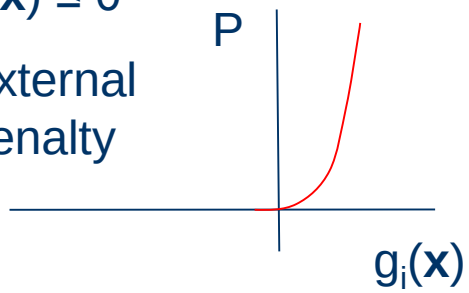


Penalty function for $h(x) = 0$

Penalty / Barrier functions

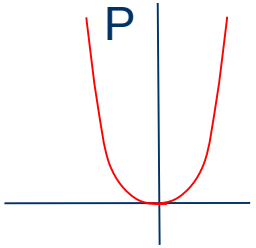
$g_i(\mathbf{x}) \leq 0$

External penalty



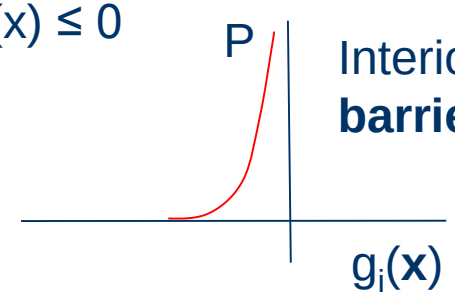
Penalty: If the constraint is violated, then P takes a large value. The optimum may violate slightly the constraint

$h(\mathbf{x}) = 0$



Parabolic penalty

$g_i(x) \leq 0$



Interior or **barrier** penalty

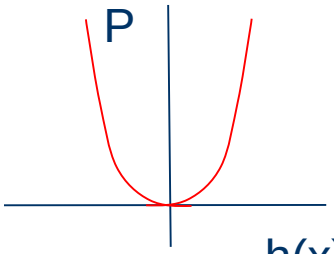
Barrier: If the value of $g_i(\mathbf{x})$ **approach** the constraint, then P takes a large value. They force x to be in the interior of the feasible set

Penalty functions

Equality constraints $h(\mathbf{x}) = 0$

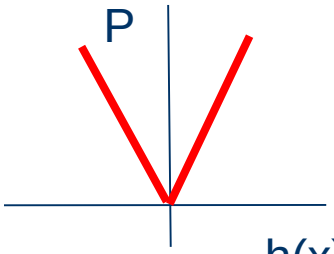
If h(**x**) deviates from the value 0, then the function P grows quadratically, penalizing the value of V(**x**) on this infeasible values. Parameter η gives the magnitude of the penalty.

If they are several equality constraints, a term $\eta(h_i(\mathbf{x}))^2$ must be added to J, for each one



Parabolic penalty $\eta h(x)^2$

P, continuous function with continuous derivatives

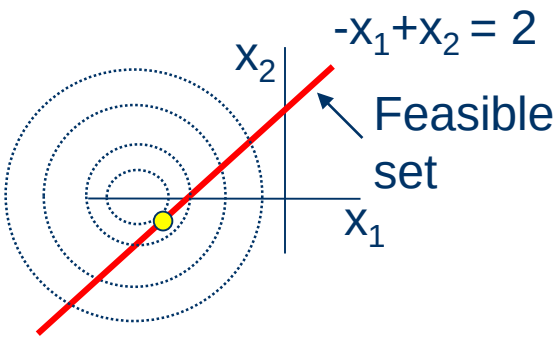


Absolute value $\eta|h(x)|$, discontinuous in h(x) = 0

Example with equality constraints

$$\left.\begin{aligned}\min_{x_1,x_2}\ & (x_1+3)^2+x_2^2\\ & -x_1+x_2-2=0\end{aligned}\right\}$$

Original problem

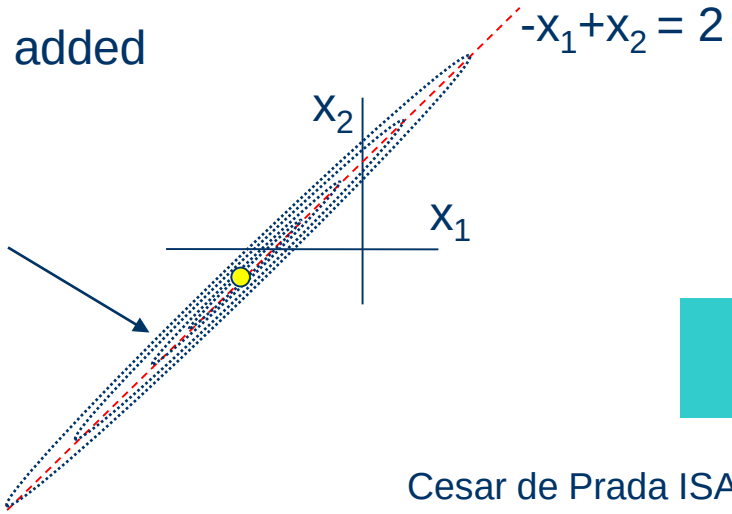


$$\min_{x_1,x_2}\ (x_1+3)^2+x_2^2+\eta(-x_1+x_2-2)^2$$

Problem with the penalty function added

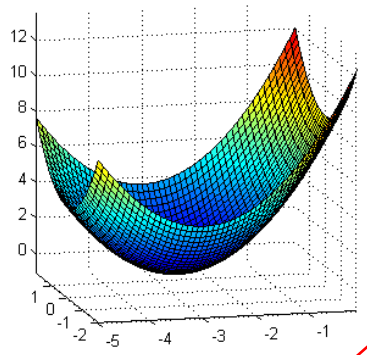
The deformed contours force to search for the minimum along the constraint $-x_1+x_2=2$

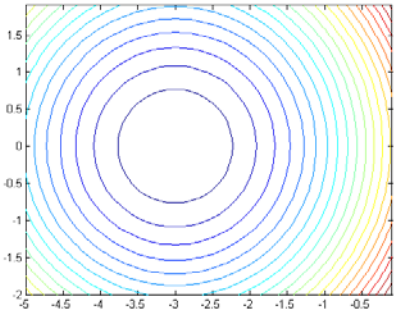
(Notice that increasing η also increases the ill-conditioning of the new problem)



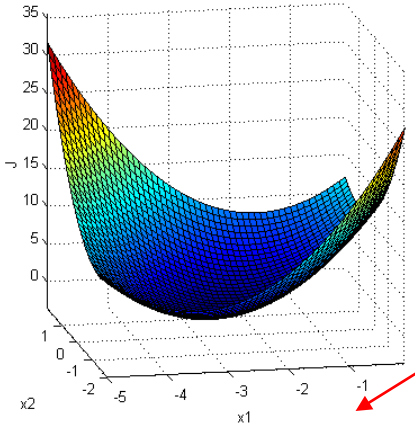
Matlab

Example 1

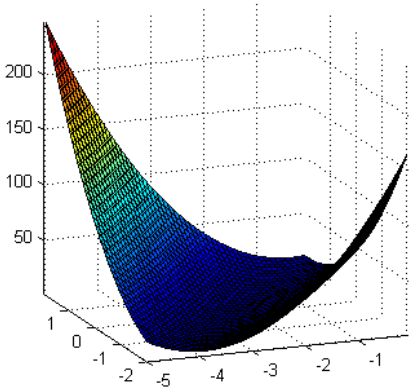


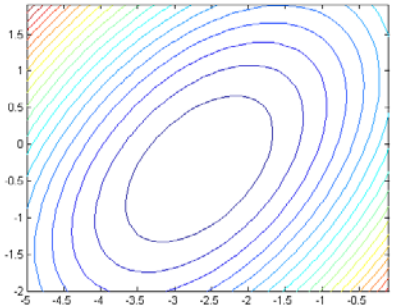


η =0 original function J(x)

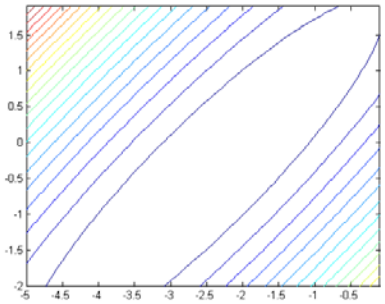


η =1 Penalty function V(x)=J(x)+ ηP





η =10 Penalty function V(x)=J(x)+ ηP



The minimum of J(x) is located around h(x)=0

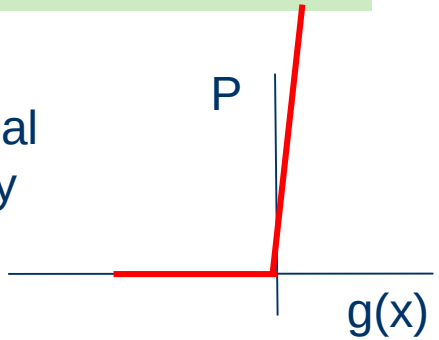
Penalty functions

Inequality constraints $g(\mathbf{x}) \leq 0$

Infinite penalty

$$P(g(\mathbf{x})) = \begin{cases} 0 & \text{if } g(\mathbf{x}) \leq 0 \\ 10^{20}|g(\mathbf{x})| & \text{if } g(\mathbf{x}) > 0 \end{cases}$$

External penalty

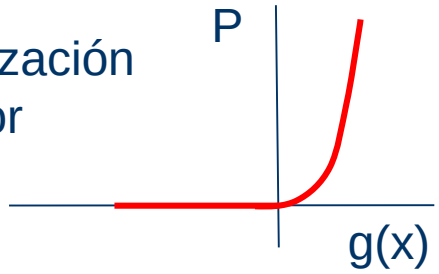


Main drawback: discontinuity at g(**x**) = 0

Bracket penalty

$$P(g(\mathbf{x})) = [\max(0, g(\mathbf{x}))]^2$$

Penalización exterior



Continuous and with continuous derivatives

As with all penalty functions, small violations of the constraints may occur

Penalty functions

Exact Penalty

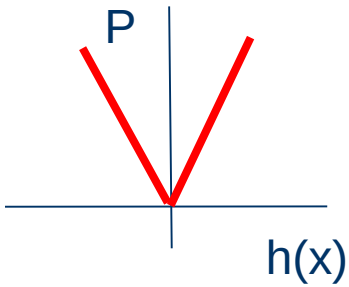
In a problem with equalities and inequalities, the minimum of the cost function:

$$\min_{\mathbf{x}} J(\mathbf{x}) + \sum_i \omega_i |h_i(\mathbf{x})| + \sum_j \sigma_i \max(0, g_j(\mathbf{x}))$$

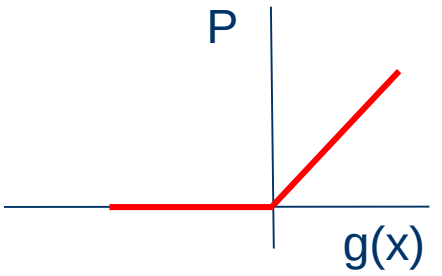
coincides exactly with the optimum x^* of the NLP problem if the weights ω_i, σ_j satisfy:

$$0 \le \omega_i \ge |\lambda_i^*| \qquad 0 \le \sigma_i \ge |\mu_i^*|$$

where x^*, λ^*, μ^* are the solution of the KKT conditions. Luenberger (1984)



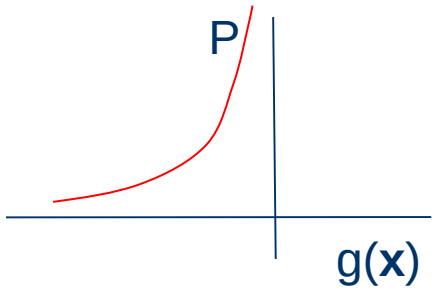
Main drawback: discontinuities at h(**x**) = 0 and g(x) = 0



Barrier functions

Inverse Barrier

$$P(g(\mathbf{x})) = -\frac{1}{g(\mathbf{x})}$$

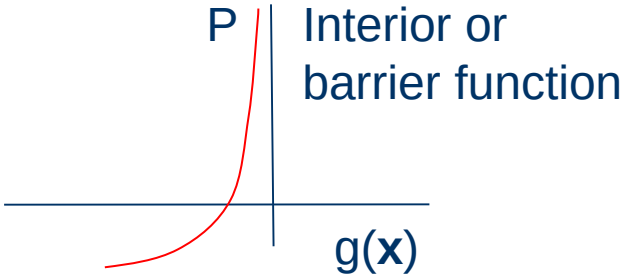


Logarithmic Barrier

$$P(g(\mathbf{x})) = -\ln(-g(\mathbf{x}))$$

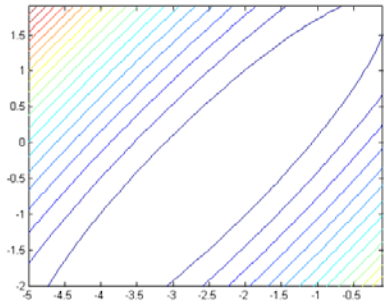
Points located in the interior of the feasible set are favoured, while those closed to the constraint g(**x**) are penalised. An infinite barrier is formed by the constraint

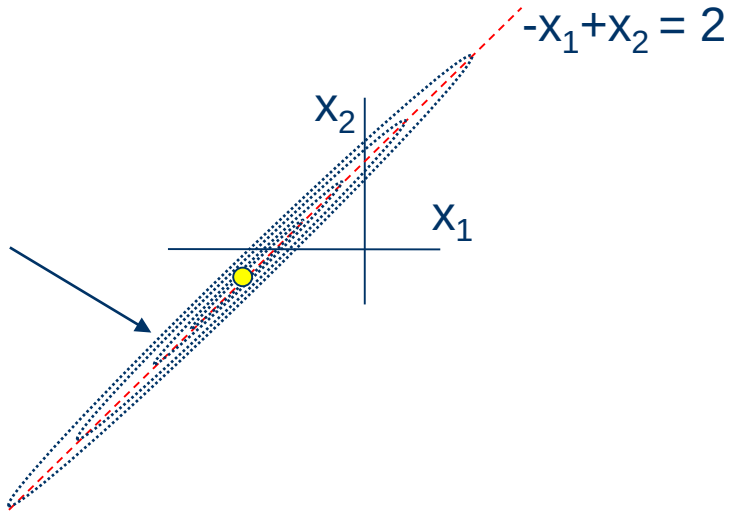
P is continuous, but if, for any reason, the constraint is violated, a difficult to recover situation is created



Notice that, with barrier function, the parameter η must be decreased progressively to allow a point to be closer to the constraints

Condition number of the Hessian





When the parameter η is modified to force the points x_k of the algorithm to be close to the constraints changing the shape of the contours, the ill-conditioning of the problem is increased. A condition number of 10^5 can be moderately large, 10^9 large and 10^{14} very large so that methods based on the inverse of the hessian will not work.

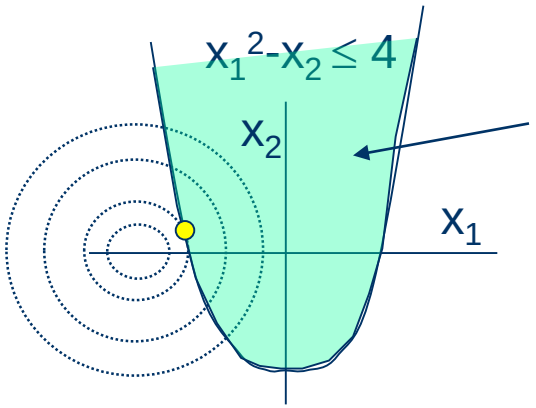
Penalty functions algorithms

1. Formulate the associated unconstraint problem with penalty (or barrier) functions
   $V(\mathbf{x},\eta) = J(\mathbf{x}) + \Sigma\eta_i P_i(g_i(\mathbf{x}),h_i(\mathbf{x}))$
2. Minimize V respect to x for a given η_i
3. Modify each η_l, increasing its value if it corresponds to an exterior penalty function or decreasing it for barrier functions
4. Check the stopping conditions and, if they are not fulfilled, return to step 2

Example with inequality constraints

$$\left.\begin{array}{l}\min\limits_{x_1,x_2}\ (x_1+3)^2+x_2^2\\ x_1^2-x_2-4\le 0\end{array}\right\}$$

Original NLP problem

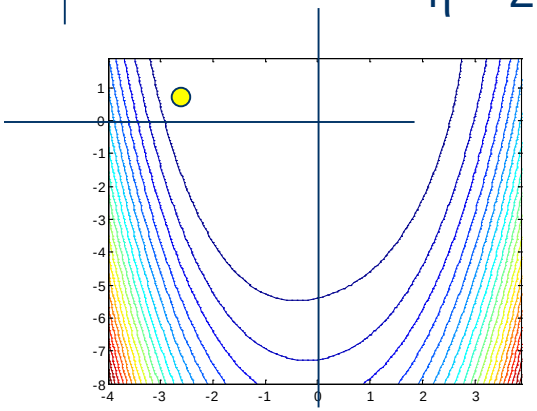


Associated problem with a quadratic penalty added

$$\min_{x_1,x_2}\ (x_1+3)^2+x_2^2+\eta P(g(x))$$

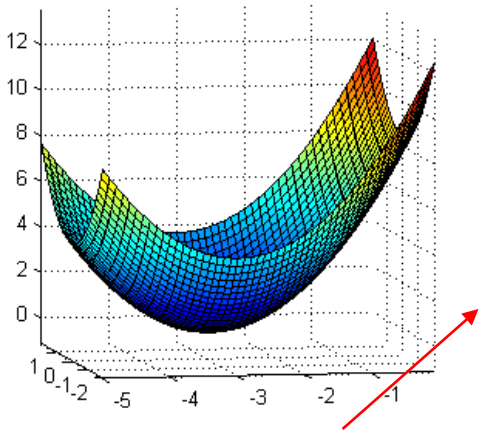
$$P(g(x))=\begin{cases}0 \ \text{ si } \ x_1^2-x_2-4\le 0\\ (x_1^2-x_2-4)^2 \ \text{ si } \ x_1^2-x_2-4>0\end{cases}$$

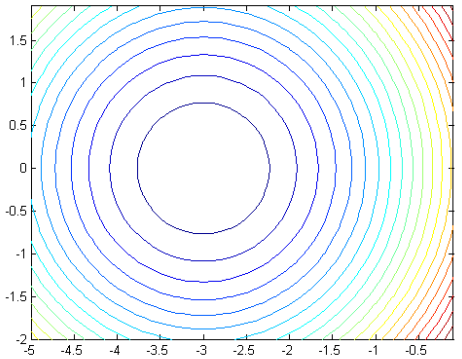
η = 2



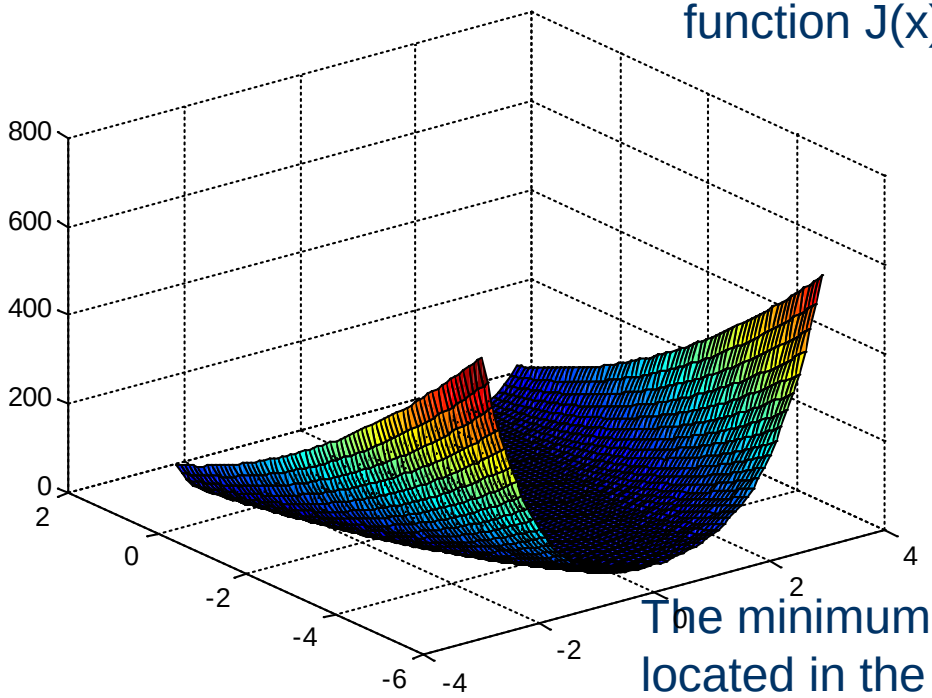
Matlab

Example 2

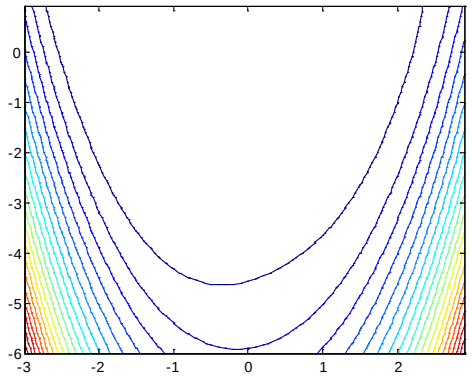




η =0 original function J(x)



η =5 penalty function $V(x)=J(x)+\eta P$

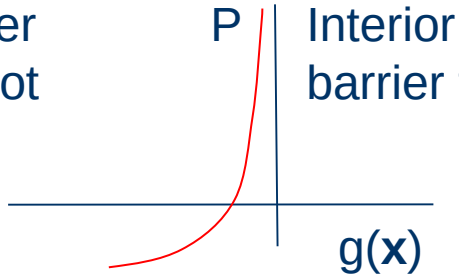


The minimum of is located in the region $g(x)\leq 0$

Barrier-penalty functions

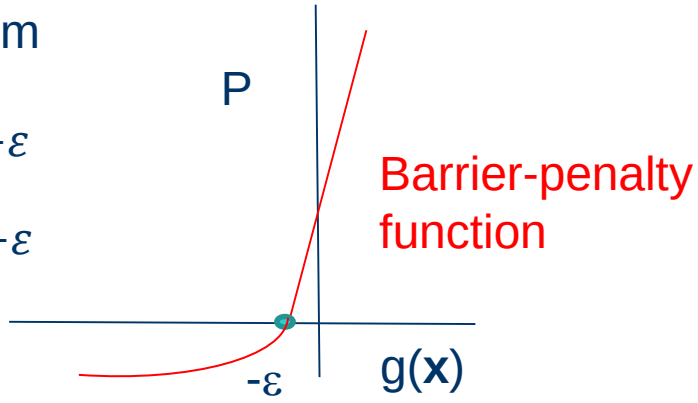
The main problem associated to the use of barrier functions as the Logarithmic Barrier is that it is not possible to recover from an infeasible point

$$P(g(\mathbf{x})) = -\ln(-g(\mathbf{x}))$$



The barrier-penalty function avoids the problem

$$\eta P\big(g(x)\big) = \begin{cases} -\eta \ln(-g(x)/\varepsilon), & g(x) \le -\varepsilon \\ \dfrac{\eta g(x)}{\varepsilon} + \eta, & g(x) > -\varepsilon \end{cases}$$



With continuous derivative.
If $\eta \to 0$ and $\varepsilon < \eta/2\lambda_{max}$ then the solution with the barrier-penalty fucntion tends to the solution of the original problem

Interior point algorithms

$$\min_{x} J(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} \leq \mathbf{0}$$

$$\min_{\mathrm{x},\varepsilon} \mathrm{J}(\mathbf{x})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} + \boldsymbol{\varepsilon} = \mathbf{0}$$
$$\boldsymbol{\varepsilon} \geq \mathbf{0}$$

$$\min_{\mathrm{x},\varepsilon} \mathrm{J}(\mathbf{x}) - \eta \sum_{\mathrm{i}=1}^{\mathrm{n}_\varepsilon} \ln(\varepsilon_{\mathrm{i}})$$
$$\mathbf{h(x)} = \mathbf{0}$$
$$\mathbf{g(x)} + \boldsymbol{\varepsilon} = \mathbf{0}$$

In interior-point solvers, the inequality constraints are handled implicitly by adding barrier terms to the cost function.

As $\eta \rightarrow 0$ the solution of the modified problem approaches the original NLP

$$\mathbf{z} = [\mathbf{x}, \boldsymbol{\varepsilon}]$$
$$\min_{\mathrm{z}} \mathrm{J}(\mathbf{z}) - \eta \sum_{\mathrm{i}=1}^{\mathrm{n}_\mathrm{z}} \ln(\mathrm{z}_{\mathrm{i}})$$
$$\mathbf{c(z)} = \mathbf{0}$$

Interior point algorithms

1. Formulate the associated equality constrained problem with barrier functions
2. $V(\mathbf{x},\eta) = J(\mathbf{x}) - \eta\Sigma \ln(\mathbf{x}_i)$, $c(x) = 0$
3. Minimize V respect to x for a given η
4. Modify η, decreasing its value towards 0
5. Check the stopping conditions and, if they are not fulfilled, return to step 2

IPOPT

$$\min_{\mathbf{x}} J(\mathbf{x}) - \eta \sum_{i=1}^{n_x} \ln(x_i)$$

$$\mathbf{c(x)} = \mathbf{0}$$

KKT

$$L = J(\mathbf{x}) - \eta \sum_{i=1}^{n_x} \ln(x_i) + \boldsymbol{\lambda}' \mathbf{c}(\mathbf{x})$$

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \eta \left[..., \frac{1}{x_i}, ... \right] = \mathbf{0}$$

$$\mathbf{c(x)} = \mathbf{0}$$

$$\mathbf{e}' = \begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x_1 & 0 & ... & 0 \\ 0 & x_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & x_n \end{bmatrix}$$

Or in matrix form (primal problem):

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \eta \mathbf{e}' X^{-1} = \mathbf{0}$$

$$\mathbf{c(x)} = \mathbf{0}$$

IPOPT

Direct numerical solution of the primal problem may be difficult due to the ill-conditioned surfaces created by the log terms. In order to solve the problem, new "dual" variables ν are created:

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \eta \mathbf{e}' \mathrm{X}^{-1} = \mathbf{0}$$

$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{v}' = \eta \mathbf{e}' \mathrm{X}^{-1}$$

$$\mathrm{X}\mathbf{v} = \eta \mathbf{e}$$

For a given η, IPOPT solves the primal-dual optimality conditions of the barrier problem directly using an exact Newton method.

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \mathbf{v}' = \mathbf{0}$$

$$\mathrm{X}\mathbf{v} = \eta \mathbf{e}$$

$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$

Interior point algorithms

$$\min_{x} J(\mathbf{x})$$
$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{x} \geq \mathbf{0}$$

KKT of the original problem

$$L = J(\mathbf{x}) + \boldsymbol{\lambda}'\mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}'\mathbf{x}$$
$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}'\nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}' = \mathbf{0}$$
$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$
$$X\boldsymbol{\nu} = \mathbf{0} \quad \boldsymbol{\nu} \geq \mathbf{0} \quad \mathbf{x} \geq \mathbf{0}$$

The primal-dual formulation can be seen as a relaxed form of the KKT conditions of the original problem, that converges to these KKT as $\eta \rightarrow 0$. Notice that x must remain > 0 and so ν according to its definition.

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}'\nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}' = \mathbf{0}$$
$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$
$$X\boldsymbol{\nu} = \eta\mathbf{e}$$

IPOPT Solution of the Primal-dual equations

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}' = \mathbf{0}$$

$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$

$$X\boldsymbol{\nu} = \eta \mathbf{e}$$

Starting with an initial guess of x, λ, ν the Newton-step is computed for a fixed η until convergence. Then η is decreased and the procedure repeated

$$\begin{bmatrix} H & \nabla_x \mathbf{c}(\mathbf{x}) & -I \\ \nabla_x \mathbf{c}(\mathbf{x})' & 0 & 0 \\ V & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta \nu \end{bmatrix} = \begin{bmatrix} \nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}' \\ \mathbf{c}(\mathbf{x}) \\ X\boldsymbol{\nu} - \eta \mathbf{e} \end{bmatrix}$$

Newton step

With H = Hessian of L
V = diag(ν)

It is possible to take advantage of the structure of the matrices to simplify the computations

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \Delta \mathbf{x}_k$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha \Delta \boldsymbol{\lambda}_k$$

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu}_k + \alpha_\nu \Delta \boldsymbol{\nu}_k$$

Stopping criterion for η

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}' = \mathbf{0}$$

$$\mathbf{c}(\mathbf{x}) = \mathbf{0}$$

$$X\boldsymbol{\nu} = \eta \mathbf{e}$$

Starting with an initial guess of x, λ, ν the Newton-step is computed for a fixed η until convergence. Then η is decreased and the procedure repeated

For the Newton step, with a fix η, the convergence criterion can be:

$$\max\left\{\left\|\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}'\right\|_\infty, \left\|\mathbf{c}(\mathbf{x})\right\|_\infty, \left\|X\boldsymbol{\nu} - \eta\mathbf{e}\right\|_\infty\right\} \leq \text{tol}$$

But notice that a similar criterion can be computed for the optimum located at η=0, so that the iterations on η can finish when:

$$\max\left\{\left\|\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \boldsymbol{\nu}'\right\|_\infty, \left\|\mathbf{c}(\mathbf{x})\right\|_\infty, \left\|X\boldsymbol{\nu}\right\|_\infty\right\} \leq \text{tol}$$

Information provided by IPOPT

Tol applies to all inf-pr, inf dual, norm X, V

Iter	Iteration number
Objetive	Original cost function
Inf_pr	Norm of constraints violation
Inf_du	The scaled dual infeasibility at the current point $\left\|\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{x}) - \mathbf{v}'\right\|_\infty$
Log(mu)	$\log_{10}$ of the value of the barrier parameter η.
|| d ||	norm of Newton step of x, ε
Lg(rg)	$\log_{10}$ of the value of the regularization term for the Hessian of the Lagrangian in the augmented system
Alpha_du	The step size for the dual variables α_v
Alpha_pr	The stepsize for the primal variables α f, h aceptation criteria (Armijo etc)
Ls	The number of backtracking line search steps

Soft and hard constraints

- Often, the constraints are classified as hard and soft constraints. The first ones are those that, as physical laws, mass balances, security limits, etc. must be fulfilled exactly. The second ones, by the contrary, may allow a certain violation of the limits, e.g. specifications, demands, etc.
- The methods that use penalty functions are very appealing when there are several constraints that can be violated at a cost.
- The problems involving penalty functions are denoted sometimes as “elastic”, because some constraint violation may occur, in opposition to the “inelastic” methods: the ones that consider hard constraints

Feasibility / Slack variables

- An alternative to the penalty functions for dealing with soft constraints, as well as a way to guarantee the existence of, at least a feasible solution, with potential infeasible LP, QP, SLP, SQP, etc., is the formulation of the problem with added slack variables in the right hand side of the constraints that must be minimized by adding them to J in an extra term

$$\min_{x,\varepsilon,\delta} J(\mathbf{x}) + \alpha\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} + \beta\boldsymbol{\delta}'\boldsymbol{\delta}$$

$$\mathbf{h(x)} = \boldsymbol{\varepsilon}$$

$$\mathbf{g(x)} \leq \boldsymbol{\delta}$$

$$\boldsymbol{\delta} \geq \mathbf{0}$$

If there is a solution to the original problem, the optimal solution of this one will give $\varepsilon = 0$, $\delta = 0$, so that it will correspond to the same solution. But if the original problem is unfeasible, ε and δ will increase the feasible region just up to the moment when a feasible solution exist