CHEM-E7225 Advanced Process Control D — Winter 2026

Lecture 03 – Nonlinear Programming (A)

Otacilio "Minho" Neto (otacilio.bneto@pm.me)

Receding-horizon policies require solving optimal control problems. The ability to formulate and solve such problems is an essential skill in advanced process control. In this lecture, we give a brief introduction to nonlinear programming; specifically, to their formulation and basic elements. Solution methods will be discussed in the next lecture.

1 Nonlinear programming

A nonlinear program is the mathematical problem stated as

$$\underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad f_0(x) \tag{1a}$$

$$\text{subject to} \quad f_i(x) = 0, \quad i = 1,\dots,N_f, \tag{1b}$$

$$h_i(x) \leq 0, \quad i = 1,\dots,N_h, \tag{1c}$$

describing the problem of finding an x (the *decision vector*) that minimizes the function f_0 (the *objective function*), while satisfying the conditions $f_i(x) = 0$ for $i = 1,\dots,N_f$ (the *equality constraints*) and $h_i(x) \leq 0$ for $i = 1,\dots,N_h$ (the *inequality constraints*). Problem (1) is said to be in *standard form*, since the right-hand side of the equality and inequality constraints are always zero. In this formulation, note that objective $f_0 : \mathbb{R}^{N_x} \to \mathbb{R}$, equality constraints $f_i : \mathbb{R}^{N_x} \to \mathbb{R}$, and inequality constraint $h_i : \mathbb{R}^{N_x} \to \mathbb{R}$ functions all scalar-valued. Hereafter, to simplify notation, we compile all the equality and inequality constraints into vector-valued functions

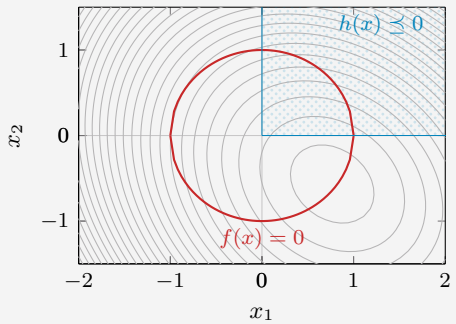
$$f(x) = \big(f_1(x), f_2(x), \dots, f_{N_f}(x)\big) = 0 \qquad \text{and} \qquad h(x) = \big(h_1(x), h_2(x), \dots, h_{N_h}(x)\big) \preceq 0 \tag{2}$$

where the symbol $\preceq$ is the generalized inequality w.r.t $\mathbb{R}^{N_x}$ (i.e., elementwise inequality). If there are no equality or inequality constraints (i.e., $N_f = N_h = 0$), the problem is said to be *unconstrained*. The nonlinear program in standard form covers all the optimization problems we care about in this course. Constraints in the form $h_l \preceq h(x) \preceq h_u$ can be written in standard form by replacing it with $h_l - h(x) \preceq 0$ and $h(x) - h_u \preceq 0$. Similarly, equality constraints in the form $f(x) = y$ can always be written as $\bar{f}(x) = f(x) - y = 0$. Finally, note that Problem (1) covers maximization problems as well, since maximizing some $\bar{f}_0(\cdot)$ is equivalent to minimizing $f_0(\cdot) = -\bar{f}_0(\cdot)$.

We say that a point $x \in \mathbb{R}^{N_x}$ is *feasible* iff it satisfies $f(x) = 0$ and $h(x) \preceq 0$. Otherwise, that point is said to be unfeasible. The *feasible set* of Problem (1) is defined as $\mathcal{X} = \big\{x \mid f(x) = 0,\ h(x) \preceq 0\big\} \subseteq \mathbb{R}^{N_x}$. If the problem has no feasible point, that is, $\mathcal{X} = \emptyset$, then the problem itself is said to be unfeasible (meaning that it cannot be solved).

Example 1.1 ***Nonlinear program in standard form***

The problem of minimizing a quadratic metric $4x_1^2 + 4x_2^2 + 2x_1x_2 - 5x_1 + 3x_2$, while satisfying the equality $x_1^2 + x_2^2 = 1$, and the inequalities $x_1 \geq 0$ and $x_2 \geq 0$, can be expressed in standard form as



$$\begin{aligned} \underset{x\in\mathbb{R}^2}{\text{minimize}} \quad & 4x_1^2 + 4x_2^2 + 2x_1x_2 - 5x_1 + 3x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 1 = 0 \\ & -x_1 \leq 0, \\ & -x_2 \leq 0 \end{aligned}$$

The plot shows the contour lines of $f_0(\cdot)$ in gray, the points satisfying $f(x) = 0$ in red, and the points satisfying $h(x) \preceq 0$ in cyan. The feasible set $\mathcal{X}$ is the intersection of the red circle with the cyan region.

1.1 Solution concepts

A solution to Problem (1) comprises two things: an *optimal value* $p^\star$ and an *optimal point* $x^\star$, formally defined by

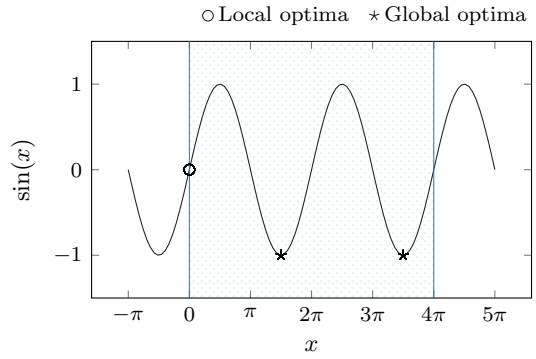
$$p^\star = \inf_x \big\{ f_0(x) \mid f(x) = 0,\ h(x) \preceq 0 \big\} \qquad \text{and} \qquad x^\star \in \arg\inf_x \big\{ f_0(x) \mid x \in f(x) = 0,\ h(x) \preceq 0 \big\}. \tag{3}$$

The latter is more important: Since $(p^\star, x^\star) = (f_0(x^\star), x^\star)$, we only need to know the optimal point to know everything else about the solution. Therefore, solving Problem (1) really means finding $x^\star$ (as stated in the beginning). We adopt the convention that $p^\star = \infty$ if $\mathcal{X} = \emptyset$ (i.e., the problem is infeasible) and that $p^\star = -\infty$ if it is feasible but unbounded below (i.e., there's always a $x' \in \mathcal{X}$ s.t. $f_0(x') \leq f_0(x)$, $x \in \mathcal{X}$). In both cases, the problem is said to be unsolvable.

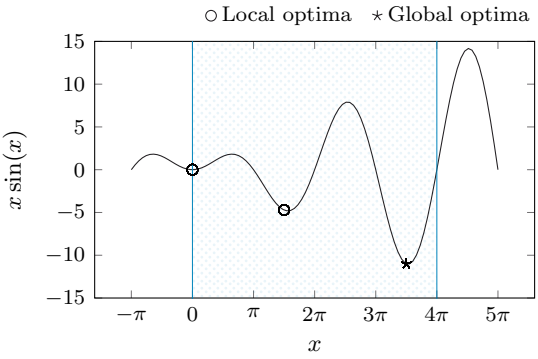
In general, the optimal point $x^\star$ is not necessarily unique, since there might be different points achieving the same optimal value $p^\star$ (Figure 1a). The *optimal/solution set* $\Omega = \big\{ x \mid f_0(x) = p^\star,\ f(x) = 0,\ h(x) \preceq 0 \big\}$ is the set of all such points. Due to practical reasons (which we'll discuss later), we sometimes have to settle for a point x that minimizes $f_0(\cdot)$ only at some local x-centered neighborhood. Formally, a *locally optimal point* x attains the *locally optimal value*

$$f(x) = \inf_z \big\{ f_0(z) \mid f(z) = 0,\ h(z) \preceq 0,\ \|x - z\|_2 \leq R \big\}, \qquad \text{given some } R > 0.$$

It is common to use the term "globally optimal" to distinguish actual solutions $(p^\star, x^\star)$ from local solutions $(f(x), x)$, as shown in Figure 1b. Unless stated otherwise, the term "optimal" in this course will always mean "globally optimal".



(a) Minimize $\sin(x)$ subject to $0 \leq x \leq 4\pi$



(b) Minimize $x\sin(x)$ subject to $0 \leq x \leq 4\pi$

Figure 1. A globally optimal point $x^\star \in \mathcal{X}$ satisfies $f(x^\star) \leq f(x)$ for all points in the feasible set, $x \in \mathcal{X}$. A locally optimal point $x^\circ \in \mathcal{X}$ satisfies $f(x^\circ) \leq f(x)$ only for the points in a R-radius neighborhood around it, $x \in \mathcal{X} \cap \{x \mid \|x^\circ - x\| \leq R\}$. In the above, Problem (a) has two global optima and one local optimum, while Problem (b) has only one global optimum and two local optima.

1.2 †Problem equivalence and transformations

Informally, two optimization problems are *equivalent* if their solutions can be related by some simple formula/routine. An important skill in optimization is learning to transform standard nonlinear programs into different formulations, which might be easier to analyze/compute. There are many such transformations (see [1, §4.1.3]). In the following, we consider two specific cases which are useful for this course.

Linear mappings into linear constraints

The following nonlinear programs are equivalent:

$$\begin{aligned} &\underset{x \in \mathbb{R}^{N_x}}{\text{minimize}} && f_0(A_0 x + b_0) \\ &\text{subject to} && h_i(A_i x + b_i) \leq 0, \quad i = 1, \dots, N_h \end{aligned} \qquad \longleftrightarrow \qquad \begin{aligned} &\underset{(x,y) \in \mathbb{R}^{N_x + N_y}}{\text{minimize}} && f_0(y_0) \\ &\text{subject to} && y_i = A_i x + b_i, \quad i = 1, \dots, N_h \\ &&& h_i(y) \leq 0, \qquad\ \ i = 1, \dots, N_h \end{aligned} \tag{4}$$

In the transformation, we augment the decision vector $(x, y) \in \mathbb{R}^{N_x + N_y}$ and introduce the linear equality constraints $y_i = A_i x + b_i$ $(i = 0, \dots, N_h)$, which allows us to rewrite the objective and constraints functions in terms of this new variable x. This might seem counterintuitive, since now the problem has more decision variables and thus should be more difficult to analyze/solve. However, it is sometimes the case that the functions (f_0, h) have some 'structure' that is computationally beneficial, but which gets lost when the decision vector undergoes a linear transformation.

Implicit to explicit constraints

The following constrained and unconstrained nonlinear programs are equivalent

$$\begin{array}{ll}\underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} & f_0(x) \\ \text{subject to} & f(x)=0,\ h(x)\preceq 0\end{array} \quad\longleftrightarrow\quad \underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad F(x) \tag{5}$$

if we define the objective function $F(\cdot)$ to contain *implicit constraints* in its definition, that is,

$$F(x)=\begin{cases} f_0(x) & \text{if } f(x)=0,\ h(x)\preceq 0 \\ \infty & \text{otherwise}\end{cases} \tag{6}$$

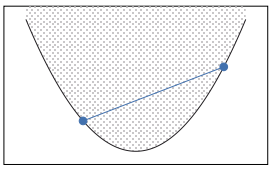
In particular, they are equivalent if we write $F(x)=f_0(x)+\sum_i I_{\{0\}}(f_i(x))+\sum_i I_{\mathbb{R}_{\leq 0}}(h_i(x))$, with $I_{\mathbb{R}_{\leq 0}}(\cdot)$ and $I_{\{0\}}(\cdot)$ being the *indicator functions* for the nonnegative reals $\mathbb{R}_{\leq 0}$ and zero-set $\{0\}$, respectively. This transformation actually brings no computational benefit (as the second problem might not even be differentiable), but it is a useful notational trick that allows us to better understand constrained optimization problems (you'll see it next lecture).

2 Convex optimization

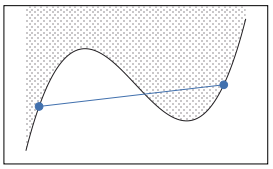
In recent history, mathematicians came to conclude that *convexity* is the single most important property in optimization. A function $f:\mathbb{R}^{N_x}\to\mathbb{R}$ is a *convex function* if its domain is a *convex set* (see [1, §2]) and it satisfies

$$f\big(\lambda x+(1-\lambda)x'\big)\leq \lambda f(x)+(1-\lambda)f(x') \tag{7}$$

for all pairs $x,x'\in\mathbb{R}^{N_x}$ and constants $0\leq\lambda\leq 1$. This important inequality can be interpreted geometrically as meaning that the line segment from $(x,f(x))$ to $(x',f(x'))$ must lie above the graph of $f(\cdot)$, as shown in Figure 2. A function f is a *concave function* if $-f$ is convex. If the inequality holds strictly (i.e., with $<$ instead of $\leq$), then f is a *strictly convex function*. If the condition holds with equality (i.e., with $=$ instead of $\leq$), then f is affine; thus, *affine functions are both convex and concave* The analysis of convex functions (and convex sets) is a well-established field which is beyond the scope of this course (read [1] for an engineering introduction to the field). Although convexity looks like a technical detail right now, we will see that it is critical in optimal control design.



(a) A convex function



(b) A nonconvex function

Figure 2

A convex program (or convex optimization problem) is the nonlinear program

$$\underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad f_0(x) \tag{8a}$$

$$\text{subject to} \quad Ax-b=0 \tag{8b}$$

$$h(x)\preceq 0 \tag{8c}$$

in which the objective function f_0 and inequality constraint functions h_i $(i=1,\dots,N_h)$ are all convex, and the equality constraints f_i $(i=1,\dots,N_f)$ are all affine. For convex programs, *local optimality implies global optimality*, that is, local information about a convex objective (i.e., its value and derivatives at a point x) provides us with global information about the problem (i.e., how distant x is from the solution). This is a remarkable property, because it allows us to design derivative-based numerical methods to (efficiently and reliably) obtain a *global* solution $(p^\star,x^\star)$. Moreover, convex programs can be categorized in *classes of problems* for which specialized *software* are built to solve.

About convex vs. nonconvex programs: Of course, convex programs are a subclass of all nonlinear programs, and it's quite common in practice to come across nonconvex problems. The practice (or rather art) of convex optimization is the ability to reformulate optimization problems as convex program. Sometimes this can be done without sacrifices, but often it requires approximating/relaxing the original problem, creating a tradeoff between *solvability* and *accuracy*.

2.1 Key properties of convex functions

In the following, $f(\cdot)$ is any arbitrary convex function whose domain **dom** f is a convex set.

The sublevels sets $C_\alpha=\{x\mid f(x)\leq\alpha\}$ are convex sets. In particular, the feasible set and ε-suboptimal sets of a convex program, $\mathcal{X}=\{x\mid Ax-b=0,\ h(x)\preceq 0\}$ and $\Omega=\big\{x\mid f_0(x)\leq p^\star+\varepsilon,\ f(x)=0,\ h(x)\preceq 0\big\}$, respectively, are convex. Thus, we say that a convex program consists of *minimizing a convex objective function over a convex set*.

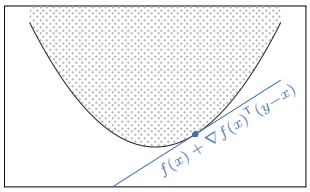


Figure 3

If f is differentiable, the *first-order condition for convexity* states that f is convex iff

$$f(y) \leq f(x) + \nabla f(x)^\mathsf{T}(y - x), \qquad \forall x, y \in \mathbf{dom}\ f. \tag{9}$$

Formally, this condition implies that the 1^{st}-order Taylor approximation of convex functions are global underestimators (Figure 3). This is perhaps the most important property: it shows that x is a *global minimizer* of f if $\nabla f(x) = 0$, since it implies that $f(y) \geq f(x)$ for all $y \in \mathbf{dom}\ f$. Thus, we can use local information (the gradient ∇f) to derive global information about f.

If f is twice-differentiable, the *second-order condition for convexity* states that f is convex iff

$$\nabla^2 f(x) \succeq 0, \qquad \forall x \in \mathbf{dom}\ f. \tag{10}$$

Geometrically, this condition implies that the graph of convex functions have positive (upward) curvature everywhere—an interesting consequence is that the negative gradient $-\nabla f(x)$ always points towards the global minimum (Figure 4). If f is strictly convex, then the Hessian is positive-definite, $\nabla^2 f(x) \succ 0$, implying that *it can always be inverted*.

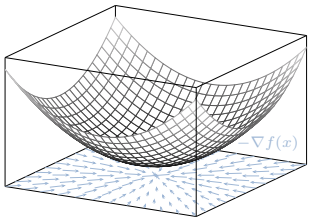


Figure 4

2.2 Composition rules and basic examples

The standard method to construct (or detect) convex function is to represent them as the composition of simpler functions whose *curvature* and *monotonicity* are known *a priori* (sometimes called *atomic functions* in some *software* libraries). The composition rules state that a function $f = h \circ g$ is convex if one of the following holds

$$h \text{ is convex and nondecreasing} \quad \text{and} \quad g \text{ is convex;} \tag{11a}$$

$$h \text{ is convex and nonincreasing} \quad \text{and} \quad -g \text{ is convex.} \tag{11b}$$

We can prove these rules using the chain rule and the second-order condition for convexity. A comprehensive list of common compositions would be too extensive for this lecture note. We will focus on a specific selection of atomic functions (Figure 5), that will be the most useful in this course.

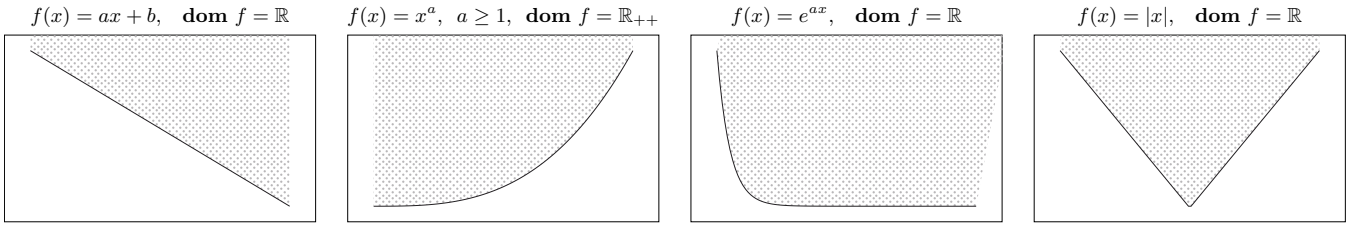


Figure 5. A selection of important (scalar) atomic convex functions.

Example 2.1 ***Convexity certification through composition rules***

The exponential function $f(x) = e^{|ax+b|}$ is a convex function, because

$$f(x) = h \circ \underbrace{h_0 \circ g_0}_{g}(x)$$

for the functions $h(z) = e^z$, $h_0(z) = |z|$, and $g_0(z) = az + b$. Since $g = h_0 \circ g_0$ is the composition of a convex function with a linear function, it is convex. Finally, since $f = h \circ g$ is the composition of a convex nondecreasing function with a convex function, it is convex.

The squared-2-norm $f(x) = \|x\|_2^2 = \sum_i x_i^2$ is convex, because

$$f(x) = \textstyle\sum_i h \circ g(x)$$

with $h(z) = z^2$ (a convex nondecreasing function) and $g(z) = 1x_i + 0$ (a linear, thus convex, function). Thus, f is convex because it is the nonnegative sum of convex functions.

3 Important classes of convex optimization problems

In advanced process control (and process systems engineering, in general), the two most fundamental classes of convex optimization problems are linear and quadratic programs.

3.1 Linear programs (LP)

A *linear program* (LP) is convex program in which all the functions are affine,

$$\begin{aligned} \underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad & c^\mathsf{T}x + d && \text{(12a)}\\ \text{subject to} \quad & Ax - b = 0, && \text{(12b)}\\ & Hx - h \preceq 0. && \text{(12c)} \end{aligned}$$

The problem data that *instantiates* an LP are the matrices (A, H), vectors (c, b, h), and scalar d. Given this data, many available solvers can efficiently tackle LPs with problem sizes $N_x \geq 10^6$ (more if the problem has *structure*).

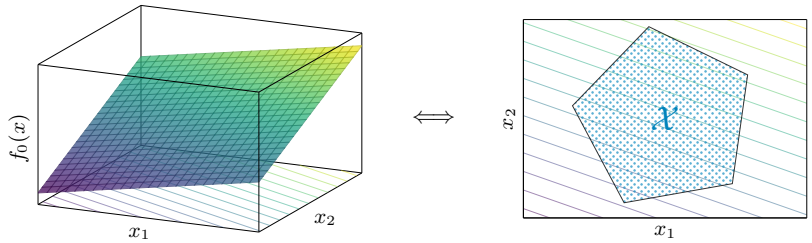


Figure 6. The cost surface f_0 (left) and level sets (right) of an LP ($N_x = 2$). Affine constraints lead to polytopic feasible sets $\mathcal{X}$.

LPs are common in maximization and scheduling problems. The *weights* $c = (c_1, \ldots, c_{N_x}) \in \mathbb{R}^{N_x}$ can be interpreted as quantifying the relative cost of each component of the decision vector $x \in \mathbb{R}^{N_x}$, with $d \in \mathbb{R}$ being a default/baseline cost. The equality constraints Eq. (12b) can be used to describe a desired distribution of x, and the inequality constraints Eq. (12c) are often bounds on the vector x (i.e., positivity constraints).

Example 3.1 ***Linear program in process systems engineering***

A isothermal CSTR of constant volume V, implements the reaction processes $A \xrightarrow{\kappa_1} B \xrightarrow{\kappa_2} C$ and $C \xrightarrow{\kappa_3}$. The chemical compound A is some reactant, C is our desired product, and B is some undesired byproduct. The feed flow-rate q [1/h] is constant, but we can manipulate the inlet concentrations $0 \leq x_{Ai} \leq 10$ [mol/L].

The problem of finding the best *steady-state operating point* can be expressed as the linear program

$$\underset{x\in\mathbb{R}^4}{\text{minimize}} \quad \begin{bmatrix} \\ c_I \\ -c_P \\ c_F \end{bmatrix}^\mathsf{T} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_{Ai} \end{bmatrix} + w_{\text{baseline}} \qquad \text{(13a)}$$

$$\text{subject to} \quad \begin{bmatrix} -(q+\kappa_1) & & & q \\ \kappa_1 & -\kappa_2 & & \\ & \kappa_2 & -\kappa_3 & \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_{Ai} \end{bmatrix} = 0, \quad \begin{bmatrix} -1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_{Ai} \end{bmatrix} - \begin{bmatrix} \\ \\ \\ \\ 10 \end{bmatrix} \preceq 0, \quad \text{(13b)}$$

with decision variables $x = (x_A, x_B, x_C, x_{Ai}) \in \mathbb{R}^4$. In the objective (13a), the weights (c_I, c_P, c_F) quantify the cost of having x_B [mols/L] of B in the final mixture, the revenue (hence, negative cost) of having x_P [mols/d] of C, and the operational cost of pumping x_{Ai} [mols/L] of reactants. The constant w_{baseline} is some baseline cost. In the constraint (13b), the equality enforces steady-state conditions (i.e., $\frac{\mathrm{d}}{\mathrm{dt}}x(t) = 0$) and the inequality enforce the decision vector to be nonnegative (i.e., $x \succeq 0$, since the decision variables represent chemical concentrations), and the operational limit in the feedstock concentrations ($x_{Ai} \leq 10$).

3.2 Quadratic programs (QP)

A *quadratic program* (QP) is a convex program in which the objective is quadratic and all constraints are affine,

$$\begin{aligned} \underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad & (1/2)x^\mathsf{T}Px + q^\mathsf{T}x + r && \text{(14a)} \\ \text{subject to} \quad & Ax - b = 0, && \text{(14b)} \\ & Hx - h \preceq 0, && \text{(14c)} \end{aligned}$$

The problem data that *instantiates* a QP are the matrices (P, A, H), vectors (q, b, h), and scalar r, all of appropriate dimensions, with the condition that $P = P^\mathsf{T} \succeq 0$ (i.e., P is a symmetric and positive-definite matrix). Note that LPs are a special case of QPs if $P = 0$. Given this data, many available solvers can solve QPs with problem sizes $N_x \geq 10^4$ in milliseconds (more if the problem has some *structure* that can be exploited).

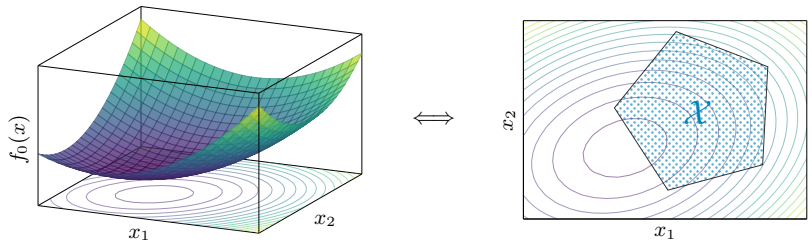


Figure 7. The cost surface f_0 (left) and level sets (right) of a QP ($N_x = 2$). Affine constraints lead to polytopic feasible sets $\mathcal{X}$.

QPs are common in control, estimation, and data fitting problems. They also appear in many subroutines of numerical algorithms to solve more complex optimization problems (including nonconvex problems). A specific instance of a QP which appear in many fields is the (unconstrained) *least-squares problem*

$$\underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad \|A_0x - b_0\|_2^2 \qquad \longleftrightarrow \qquad \underset{x\in\mathbb{R}^{N_x}}{\text{minimize}} \quad (1/2)x^\mathsf{T}Px + q^\mathsf{T}x + r \tag{15}$$

with $P = 2A_0^\mathsf{T}A_0$, $q = -2b_0^\mathsf{T}A_0$ and $r = b_0^\mathsf{T}b_0$. This class of problems have the well-known analytical solution $x^\star = A_0^\dagger b_0$, with $A_0^\dagger$ being the pseudo-inverse of A_0. If constraints are present, we don't have an analytical solution but the problem is still convex (and thus can be solved efficiently). Least-squares problems are so ubiquitous and fundamental, that modern computer systems already come with some built-in low-level routines to solve them.

Example 3.2 ***Least-squares program in process systems engineering***

In a pulping process, we want to predict the depth x [mm] of cooking liquor infiltration in a woodchip. We propose that this infiltration follows the Nth-order AutoRegressive, AR(N), model

$$x[n] = \sum_{\tau=1}^{N} \alpha_\tau x[n-\tau] = \alpha_1 x[n-1] + \cdots + \alpha_{N-1}x[n-N-1] \tag{16}$$

However, the specific values of the model parameters $\alpha = (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N$ are unknown. We perform a laboratory experiment and measure the infiltration in a single woodchip, obtaining the data

$$\begin{matrix} n \\ x[n] \end{matrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 36 & 56 & 68 & 75 & 79 & 81 & 82 & 83 \end{bmatrix}. \tag{17}$$

The parameters that best fit the data (in a least-squares sense) can be obtained from the problem

$$\underset{\alpha\in\mathbb{R}^N}{\text{minimize}} \quad \textstyle\sum_{n=0}^{8} \left(x[n] - \sum_{\tau=1}^{N} \alpha_\tau x[n-\tau]\right)^2, \tag{18}$$

which can be converted into a matrix-form QP after some algebra. When fitting autoregressive models, it is recommended to subtract the steady-state value (here, note that $x[n] \to 83$), such that the AR(N) is actually modelling deviations around the equilibrium point.

References

[1] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. ISBN: 978-0-52-183378-3.