CHAPTER 2: QUADRATIC PROGRAMMING

Overview

Quadratic programming (QP) problems are characterized by objective functions that are quadratic in the design variables, and linear constraints. In this sense, QPs are a generalization of LPs and a special case of the general nonlinear programming problem. QPs are ubiquitous in engineering problems, include civil & environmental engineering systems. A classic example is least squares optimization, often performed during regression analysis. Like LPs, QPs can be solved graphically. However, the nature of solutions is quite different. Namely, interior optima are possible. This leads us towards conditions for optimality, which is an extension of basic optimization principles learned in first-year calculus courses. Finally, QPs provide a building-block to approximately solve nonlinear programs. That is, a nonlinear program can be solved by appropriately constructing a sequence of approximate QPs.

By the end of this chapter, students will be able to identify and formulate QPs. They will also be able to assess the nature of its solution, i.e. unique local maxima/minima, infinite solutions, or no solutions. Finally, they will have one tool to approximately solve a more general nonlinear programming problem.

Chapter Organization

This chapter is organized as follows:

- (Section 1) Quadratic Programs (QP)
- (Section 2) Least Squares
- (Section 3) Graphical QP
- (Section 4) Optimality Conditions
- (Section 5) Sequential Quadratic Programming (SQP)

1 Quadratic Programs

A quadratic program (QP) is the problem of optimizing a quadratic objective function subject to linear constraints. Mathematically,

$$\text{Minimize:} \qquad \frac{1}{2}x^T Q x + R^T x + S \tag{1}$$

$$\text{subject to:} \quad Ax \leq b \tag{2}$$

$$A_{eq}x = b_{eq} \tag{3}$$

where $x \in \mathbb{R}^n$ is the vector of design variables. The remaining matrices/vectors have dimensions $Q \in R^{n \times n}$, $R \in \mathbb{R}^n$, $S \in \mathbb{R}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $A_{eq} \in \mathbb{R}^{l \times n}$, $b_{eq} \in \mathbb{R}^l$, where n is the number of design variables, m is the number of inequality constraints, and l is the number of equality constraints.

Remark 1.1 (The "S" Term)**.** Note that the S term in (1) can be dropped without loss of generality, since it has no impact on the optimal solution x^*. That is, suppose x^* is the optimal solution to (1)-(3). Then it is also the optimal solution to

$$\text{Minimize:} \quad \frac{1}{2}x^T Qx + R^T x \tag{4}$$

$$\text{subject to:} \quad Ax \leq b \tag{5}$$

$$A_{eq}x = b_{eq} \tag{6}$$

As a consequence, we often disregard the S term. In fact, some computational solvers do not even consider the S term as an input.

Remark 1.2 (Quadratically Constrained Quadratic Program)**.** A generalization of the QP is to consider quadratic inequality constraints, resulting in the quadratically constrained quadratic program (QCQP)

$$\text{Minimize:} \quad \frac{1}{2}x^T Qx + R^T x \tag{7}$$

$$\text{subject to:} \quad \frac{1}{2}x^T Ux + V^T x + W \leq 0 \tag{8}$$

$$A_{eq}x = b_{eq} \tag{9}$$

When $U = 0$, then this problem degenerates into a standard QP. We will not discuss this class of QPs further, besides to say it exists along with associated solvers.

2 Least Squares

To this point, QPs are simply an abstract generalization of LPs. In this section, we seek to provide some practical motivation for QPs. Consider an overdetermined linear system of equations. That is consider a "skinny" matrix A, and the following equation to solve:

$$y = Ax \tag{10}$$

where $x \in \mathbb{R}^n$ is a vector of unknowns, $y \in \mathbb{R}^m$ includes known data, and $A \in \mathbb{R}^{m \times n}$ is a known matrix where $m > n$. Under appropriate conditions[1] for A, eqn. (10) has no unique solution, since there are more equations than unknowns. We call this an "overdetermined" systems of equations.

A common approach is to seek a "best fit" solution. For this, define the residual error as

$$r = Ax - y \tag{11}$$

and consider a solution x^* that minimizes the norm of the residual, $\|r\|$. This solution is called the "least squares" solution. It is often used in regression analysis, among many applications, as demonstrated below.

Example 2.1 (Linear Regression)**.** Suppose you have collected measured data pairs (x_i, y_i), for $i = 1, \cdots, N$ where $N > 6$, as shown in Fig. 1. You seek to fit a fifth-order polynomial to this data, i.e.

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 \tag{12}$$

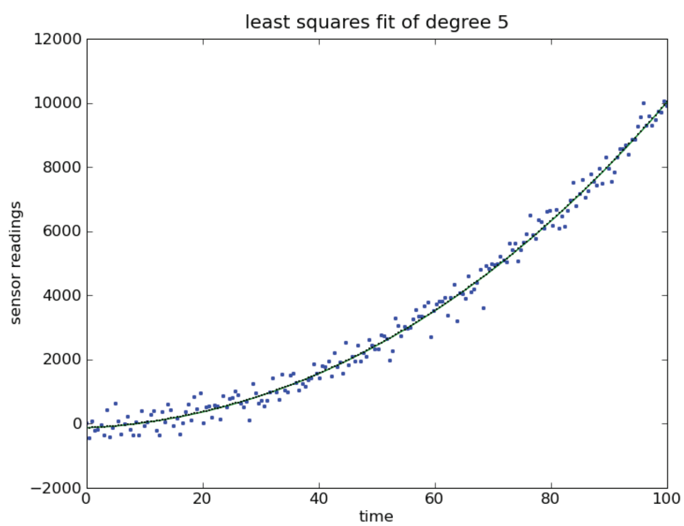


Figure 1: You seek to fit a fifth-order polynomial to the measured data above.

The goal is to determine parameters $c_j,\ j = 0, \cdots, 5$ that "best" fit the data in some sense. To this end, you may compute the residual r for each data pair:

$$\begin{aligned} c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_1^3 + c_4 x_1^4 + c_5 x_1^5 - y_1 &= r_1, \\ c_0 + c_1 x_2 + c_2 x_2^2 + c_3 x_2^3 + c_4 x_2^4 + c_5 x_2^5 - y_2 &= r_2, \\ \vdots &= \vdots \\ c_0 + c_1 x_N + c_2 x_N^2 + c_3 x_N^3 + c_4 x_N^4 + c_5 x_N^5 - y_N &= r_N, \end{aligned} \tag{13}$$

[1] All columns of A must be linearly independent.

which can be arranged into matrix-vector form $Ac - y = r$, where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 & x_N^4 & x_N^5 \end{bmatrix}, \qquad c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}, \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}. \tag{14}$$

Now we compute an optimal fit for c in the following sense. We seek the value of c which minimizes the squared residual

$$\min_c \quad \frac{1}{2}\|r\|^2 = \frac{1}{2}r^T r = \frac{1}{2}(Ac-y)^T(Ac-y) = \frac{1}{2}c^T A^T A c - y^T A c + \frac{1}{2}y^T y. \tag{15}$$

Note that (15) is quadratic in variable c. In this case the problem is also unconstrained. As a result, we can set the gradient with respect to c to zero and directly solve for the minimizer.

$$\begin{aligned} \frac{\partial}{\partial c}\frac{1}{2}\|r\|^2 = A^T A c - A^T y &= 0, \\ A^T A c &= A^T y, \\ \boxed{c = (A^T A)^{-1} A^T y} \end{aligned} \tag{16}$$

This provides a direct formula for fitting the polynomial coefficients $c_j,\ j = 0, \cdots, 5$ using the measured data.

Exercise 1. *Consider fitting the coefficients c_1, c_2, c_3 of the following sum of radial basis functions to data pairs (x_i, y_i), $i = 1, \cdots, N$.*

$$y = c_1 e^{-(x-0.25)^2} + c_2 e^{-(x-0.5)^2} + c_3 e^{-(x-0.75)^2} \tag{17}$$

Formulate and solve the corresponding QP problem.

Exercise 2. *Repeat the same exercise for the following Fourier Series:*

$$y = c_1 \sin(\omega x) + c_2 \cos(\omega x) + c_3 \sin(2\omega x) + c_4 \cos(2\omega x) \tag{18}$$

3 Graphical QP

For problems of one, two, or three dimensions, it is possible to solve QPs graphically. Consider the following QP example:

$$\min \qquad J = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\begin{aligned} \text{s. to} \qquad 2x_1 + 4x_2 &\leq 28 \\ 5x_1 + 5x_2 &\leq 50 \\ x_1 &\leq 8 \\ x_2 &\leq 6 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

The feasible set and corresponding iso-contours are illustrated in the left-hand side of Fig. 2. In this case, the solution is an **interior optimum.** That is, no constraints are active at the minimum. In contrast, consider the objective function $J = (x_1 - 6)^2 + (x_2 - 6)^2$ shown on the right-hand side of Fig. 2. In this case, the minimum occurs at the boundary and is unique.

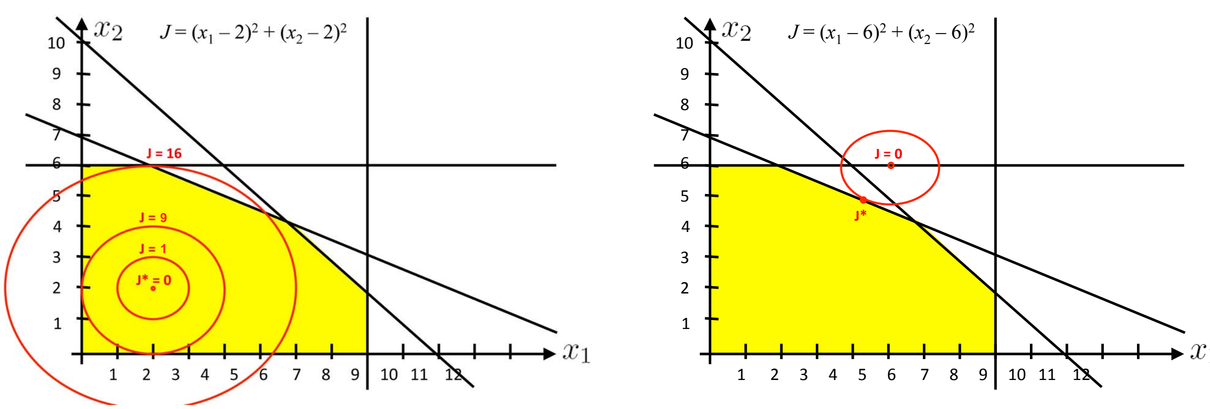


Figure 2: An interior optimum [LEFT] and boundary optimum [RIGHT] for a QP solved graphically.

Exercise 3. *Solve the following QP graphically*

$$\min \qquad J = (x_1 - 2)^2 + (x_2 - 4)^2$$

$$\begin{array}{lrcl} \textit{s. to} & 5x_1 + 3x_2 & \leq & 15 \\ & x_1 & \geq & 3 \\ & x_2 & \leq & 0 \end{array}$$

Exercise 4. *Solve the following QP graphically*

$$\min \qquad J = x_2^2$$

$$\begin{array}{lrcl} \textit{s. to} & x_1 + x_2 & \leq & 10 \\ & x_1 & \geq & 0 \\ & x_1 & \leq & 5 \end{array}$$

4 Optimality Conditions

In this section, we establish a connection to optimization from high-school / first-year college[2] calculus. We establish this connection within the context of QPs, since they also reveal the nature of QP solutions.

Consider an unconstraint QP of the form

$$\min \quad f(x) = \frac{1}{2} x^T Q x + R^T x \tag{19}$$

In calculus, you learned that a necessary condition for minimizers is that the function's slope is zero at the optimum. We extend this notion to multivariable functions. That is, if x^* is an optimum, then the gradient is zero at the optimum. Mathematically,

$$\begin{aligned} \frac{d}{dx} f(x^*) &= 0 \\ &= Qx^* + R \\ \Rightarrow Qx^* &= -R \end{aligned} \tag{20}$$

We call this condition the first order necessary condition (FONC) for optimality. This condition is necessary for an optimum, but not sufficient for completely characterizing a minimizer or maximizer. As a result, we call a solution to the FONC a stationary point, $x^\dagger$. In calculus, the term "extremum" is often used.

Recall from calculus that the second derivative can reveal if a stationary point is a minimizer, maximizer, or neither. For single variable functions $f(x)$, the stationary point $x^\dagger$ has nature char-

[2] At UC Berkeley, the related course in Math 1A

acterized by the following[3].

$$\begin{aligned} x^\dagger &\text{ is minimizer} \quad \text{if} \quad f''(x^\dagger) > 0 \\ x^\dagger &\text{ is maximizer} \quad \text{if} \quad f''(x^\dagger) < 0 \\ x^\dagger &\text{ is inflection point} \quad \text{if} \quad f''(x^\dagger) = 0 \end{aligned} \tag{21}$$

We now extend this notion to multivariable optimization problems. Consider the second derivative of a multivariable function, which is called the Hessian. Mathematically,

$$\frac{d^2}{dx^2} f(x^\dagger) = Q \tag{22}$$

and Q is the Hessian. The nature of the stationary point is given by the positive definiteness of matrix Q, as shown in Table 1. Namely, if Q is positive definite then $x^\dagger$ is a local minimizer. If Q is negative definite then $x^\dagger$ is a local maximizer. It is also possible to have infinite solutions, which can be characterized by the Hessian Q. If Q is positive (negative) semi-definite, then $x^\dagger$ is a valley (ridge). If Q is indefinite, then $x^\dagger$ is a saddle point. We call this the second order sufficient condition (SOSC). Visualizations of each type of stationary point are provided in Fig. 3 and 4.

A quick review of positive definite matrices is provided in Section 4.1.

Table 1: Relationship between Hessian and nature of stationary point $x^\dagger$.

Hessian matrix	**Quadratic form**	**Nature of $x^\dagger$**
positive definite	$x^T Q x > 0$	local minimizer
negative definite	$x^T Q x < 0$	local maximizer
positive semi-definite	$x^T Q x \geq 0$	valley
negative semi-definite	$x^T Q x \leq 0$	ridge
indefinite	$x^T Q x$ any sign	saddle point

[3]Technically, an inflection point is a point $x^\dagger$ where the curve $f(x)$ changes from concave to convex, or vice versa. It is possible for $f''(x^\dagger) = 0$ and the concavity does not change. An example is $f(x) = x^4$ for $x^\dagger = 0$. In this case $x^\dagger$ is not an inflection point, but undulation point.

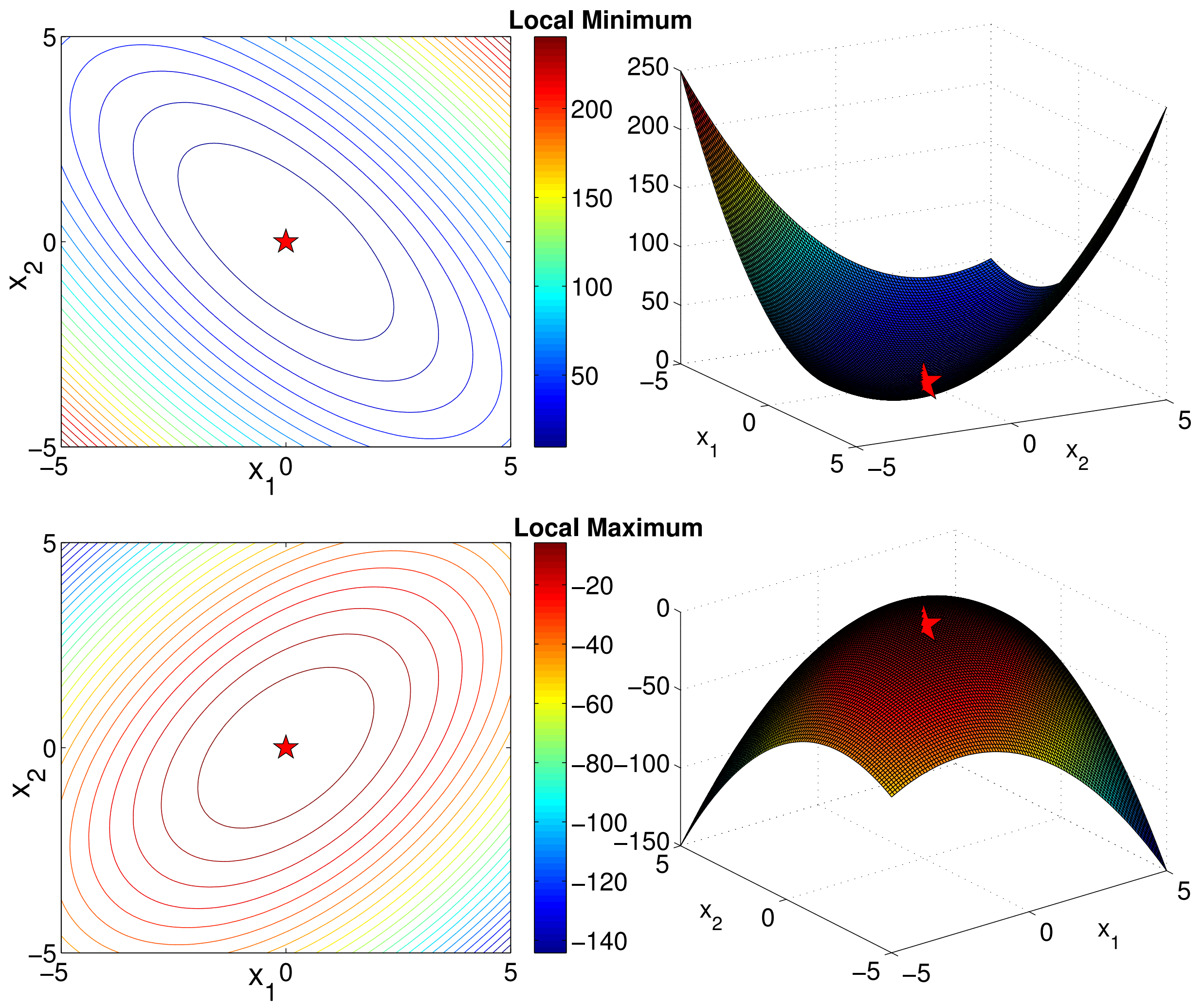


Figure 3: Visualizations of stationary points of different nature. In each case, the objective functions take the form $f(x_1, x_2) = x^T Q x$ and have stationary points at the origin $(x_1^\dagger, x_2^\dagger) = (0, 0)$, denoted by the red star. The Hessian for each case are: Local Minimum $Q = [3\ 2;\ 2\ 3]$, Local Maximum $Q = [-2\ 1;\ 1\ -2]$.

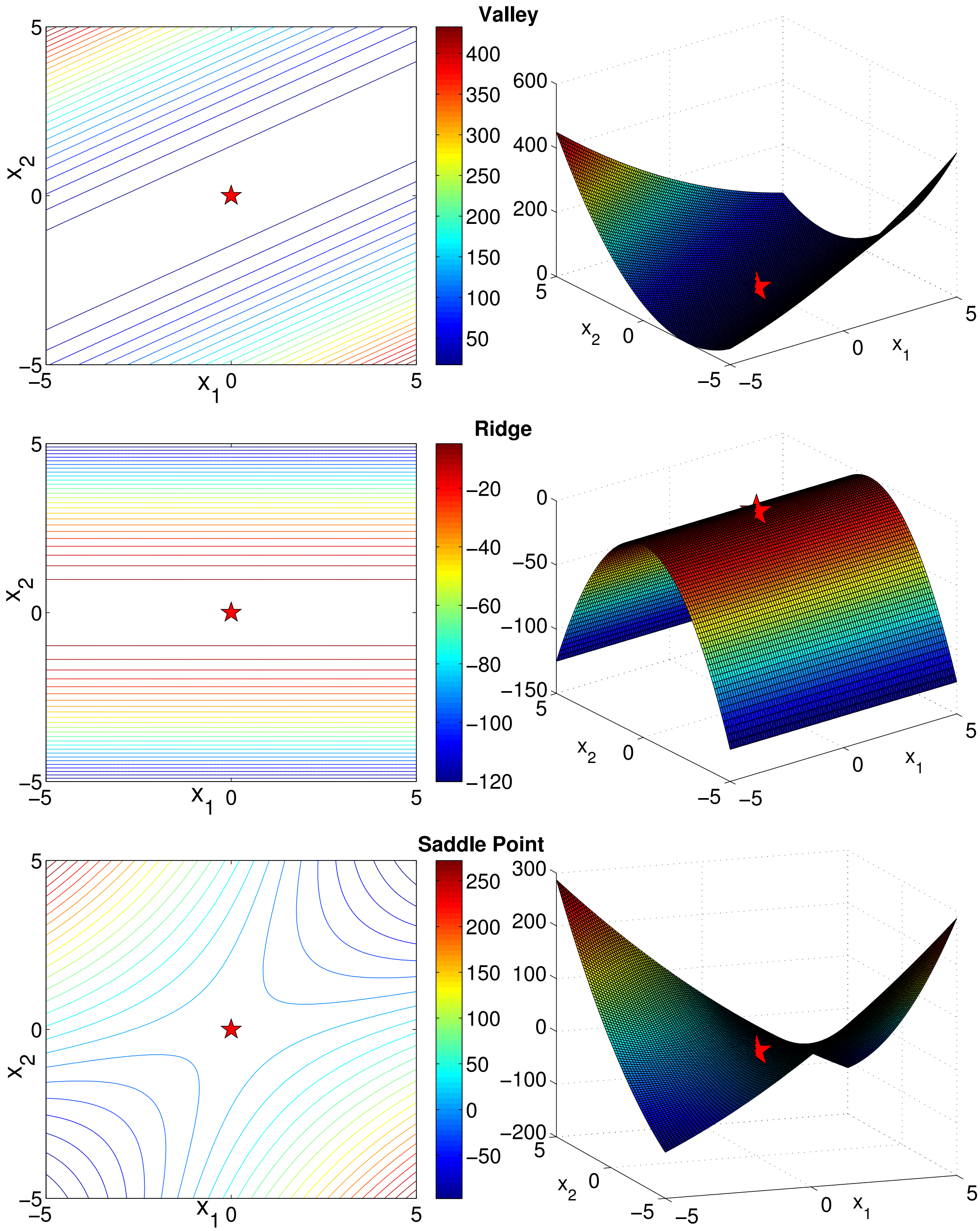


Figure 4: Visualizations of stationary points of different nature. In each case, the objective functions take the form $f(x_1, x_2) = x^T Q x$ and have stationary points at the origin $(x_1^\dagger, x_2^\dagger) = (0, 0)$, denoted by the red star. The Hessian for each case are: Valley $Q = [8\ -4;\ -4\ 2]$, Ridge $Q = [-5\ 0;\ 0\ 0]$, Saddle Point $Q = [2\ -4;\ -4\ 1.5]$.

4.1 Review of Positive Definite Matrices

In linear algebra, matrix positive definiteness is a generalization of positivity for scalar variables.

Definition 4.1 (Positive Definite Matrix)**.** *Consider symmetric matrix $Q \in \mathbb{R}^{n \times n}$. All of the following conditions are equivalent:*

- Q *is positive definite*
- $x^T Q x > 0, \forall\, x \neq 0$
- *the real parts of all eigenvalues of* Q *are positive*
- $-Q$ *is negative definite*

Similarly, the following conditions are all equivalent:

- Q *is positive semi-definite*
- $x^T Q x \geq 0, \forall\, x \neq 0$
- *the real parts of all eigenvalues of* Q *are positive, and at least one eigenvalue is zero*
- $-Q$ *is negative semi-definite*

In practice, the simplest way to check positive-definiteness is to examine the signs of the eigenvalues. In Matlab, one can compute the eigenvalues using the `eig` command.

Exercise 5. *Determine if the following matrices are positive definite, negative definite, positive semi-definite, negative semi-definite, or indefinite.*

(a) $M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(b) $M = \begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}$

(c) $M = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

(d) $M = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

(e) $M = \begin{bmatrix} -8 & 4 \\ 4 & -2 \end{bmatrix}$

(f) $M = \begin{bmatrix} 4 & -2 & 0 \\ -2 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}$

4.2 Examples for Optimality Conditions

Example 4.1. Consider the following unconstrained QP

$$\begin{aligned} f(x_1, x_2) &= (3 - x_1)^2 + (4 - x_2)^2 \\ &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -6 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{23}$$

Check the FONC. That is, find values of $x = [x_1, x_2]^T$ where the gradient of $f(x_1, x_2)$ equals zero.

$$\frac{\partial f}{\partial x} = \begin{bmatrix} -6 + 2x_1 \\ -8 + 2x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{24}$$

has the solution $(x_1^\dagger, x_2^\dagger) = (3, 4)$. Next, check the SOSC. That is, check the positive definiteness of the Hessian.

$$\frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \rightarrow \text{positive definite} \tag{25}$$

since the eigenvalues are 2,2. Consequently, the point $(x_1^*, x_2^*) = (3, 4)$ is a unique minimizer.

Example 4.2. Consider the following unconstrained QP

$$\begin{aligned} \min_{x_1, x_2} f(x_1, x_2) &= -4x_1 + 2x_2 + 4x_1^2 - 4x_1 x_2 + x_2^2 \\ &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 4 & -2 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{26}$$

Check the FONC. That is, find values of $x = [x_1, x_2]^T$ where the gradient of $f(x_1, x_2)$ equals zero.

$$\frac{\partial f}{\partial x} = \begin{bmatrix} -4 + 8x_1 - 4x_2 \\ 2 - 4x_1 + 2x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{27}$$

has an infinity of solutions $(x_1^\dagger, x_2^\dagger)$ on the line $2x_1 - x_2 = 1$. Next, check the SOSC. That is, check the positive definiteness of the Hessian.

$$\frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} 8 & -4 \\ -4 & 2 \end{bmatrix} \rightarrow \text{positive semidefinite} \tag{28}$$

since the eigenvalues are 0,10. Consequently, there is an infinite set of minima (valley) on the line $2x_1^* - x_2^* = 1$.

Exercise 6. *Examine the FONC and SOSC for the following unconstrained QP problems. What is the stationary point $x^\dagger$? What is its nature, i.e. unique minimizer, unique maximizer, valley, ridge, or no solution?*

(a) $\min_{x_1,x_2} \;\; f(x_1,x_2) = x_1^2 - 3x_1x_2 + 4x_2^2 + x_1 - x_2$

(b) $\min_{x_1,x_2} \;\; f(x_1,x_2) = 2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2$

(c) $\min_{x_1,x_2,x_3} \;\; f(x_1,x_2,x_3) = x_1^2 + 2x_2^2 + 3x_3^2 + 3x_1x_2 + 4x_1x_3 - 3x_2x_3$

(d) $\min_{x_1,x_2,x_3} \;\; f(x_1,x_2,x_3) = 2x_1^2 + x_1x_2 + x_2^2 + x_2x_3 + x_3^2 - 6x_1 - 7x_2 - 8x_3 + 19$

(e) $\min_{x_1,x_2,x_3} \;\; f(x_1,x_2,x_3) = x_1^2 + 4x_2^2 + 4x_3^2 + 4x_1x_2 + 4x_1x_3 + 16x_2x_3$

5 Sequential Quadratic Programming

In our discussion of QPs so far, we have defined QPs, motivated their use with least squares in regression analysis, examined graphical solutions, and discussed optimality conditions for unconstrained QPs. To this point, however, we are still not equipped to solve general nonlinear programs (NLPs). In this section, we provide a direct method for handling NLPs with constraints, called the Sequential Quadratic Programming (SQP) method. The idea is simple. We solve a single NLP as a sequence QP subproblems. In particular, at each iteration we approximate the objective function and constraints by a QP. Then, within each iteration, we solve the corresponding QP and use the solution as the next iterate. This process continues until an appropriate stopping criterion is satisfied.

SQP is very widely used in engineering problems and often the first "go-to" method for NLPs. For many practical energy system problems, it produces fast convergence thanks to its strong theoretical basis. This method is commonly used under-the-hood of Matlab function `fmincon`.

Consider the general NLP

$$\min_{x} \quad f(x) \tag{29}$$

$$\text{subject to} \quad g(x) \leq 0, \tag{30}$$

$$h(x) = 0, \tag{31}$$

and the k^{th} iterate x_k for the decision variable. We utilize the Taylor series expansion. At each iteration of SQP, we consider the 2nd-order Taylor series expansion of the objective function (29), and 1st-order expansion of the constraints (30)-(31) around $x = x_k$:

$$f(x) \approx f(x_k) + \frac{\partial f}{\partial x}^T (x_k)\,(x - x_k) + \frac{1}{2}(x - x_k)^T \frac{\partial^2 f}{\partial x^2}(x_k)\,(x - x_k), \tag{32}$$

$$g(x) \approx g(x_k) + \frac{\partial g}{\partial x}^T (x_k)\,(x - x_k) \leq 0, \tag{33}$$

$$h(x) \approx h(x_k) + \frac{\partial h}{\partial x}^T (x_k)\,(x - x_k) = 0. \tag{34}$$

To simplify the notation, define $\tilde{x} = x - x_k$. Then we arrive at the following approximate QP

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x}, \tag{35}$$

$$\text{s. to} \quad A\tilde{x} \leq b \tag{36}$$

$$A_{eq}\tilde{x} = b_{eq} \tag{37}$$

where

$$Q = \frac{d^2 f}{dx^2}(x_k), \qquad R = \frac{df}{dx}(x_k), \tag{38}$$

$$A = \frac{dg}{dx}^T (x_k), \qquad b = -g(x_k), \tag{39}$$

$$A_{eq} = \frac{dh}{dx}^T (x_k), \qquad b_{eq} = -h(x_k). \tag{40}$$

Suppose (35)-(37) yields the optimal solution $\tilde{x}^*$. Then let $x_{k+1} = x_k + \tilde{x}^*$, and repeat.

Remark 5.1. Note that the iterates in SQP are not guaranteed to be feasible for the original NLP problem. That is, it is possible to obtain a solution to the QP subproblem which satisfies the approximate QP's constraints, but not the original NLP constraints.

Example 5.1. *Consider the NLP*

$$\min_{x_1, x_2} \quad e^{-x_1} + (x_2 - 2)^2, \tag{41}$$

$$\textit{s. to} \quad x_1 x_2 \leq 1. \tag{42}$$

with the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 1]^T$*. By hand, formulate the Q,R,A,b matrices for the first three iterates. Use Matlab command* `quadprog` *to solve each subproblem. What is the solution after three iterations?*

We have $f(x) = e^{-x_1} + (x_2 - 2)^2$ and $g(x) = x_1 x_2 - 1$. The iso-contours for the objective function and constraint are provided in Fig. 5. From visual inspection, it is clear the optimal solution is near

$[0.5, 2]^T$. We seek to find the approximate QP subproblem

$$\min \quad \frac{1}{2}\tilde{x}^T Q \tilde{x} + R^T \tilde{x} \tag{43}$$

$$\text{s. to} \quad A\tilde{x} \leq b \tag{44}$$

Taking derivatives of $f(x)$ and $g(x)$, one obtains

$$Q = \begin{bmatrix} e^{-x_1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-x_1} \\ 2(x_2 - 2) \end{bmatrix}, \tag{45}$$

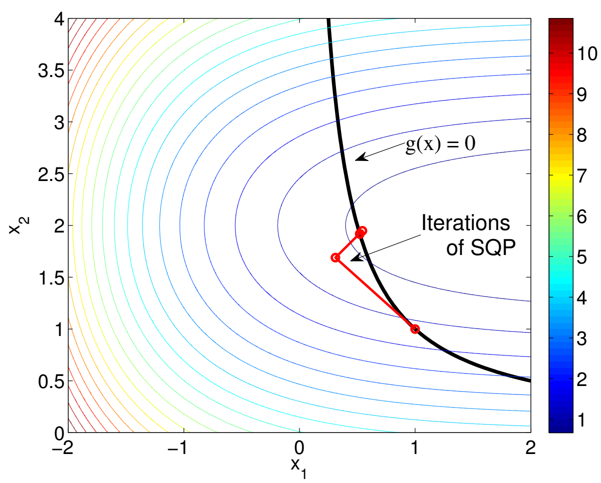
$$A = \begin{bmatrix} x_2 & x_1 \end{bmatrix}, \qquad b = 1 - x_1 x_2 \tag{46}$$

Now consider the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 1]^T$. Note that this guess is feasible. We obtain the following matrices for the first QP subproblem

$$Q = \begin{bmatrix} e^{-1} & 0 \\ 0 & 2 \end{bmatrix}, \qquad R = \begin{bmatrix} -e^{-1} \\ -2 \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad b = 0$$

Solving this QP subproblem results in $\tilde{x}^* = [-0.6893, 0.6893]$. Then the next iterate is given by $[x_{1,1}, x_{2,1}] = [x_{1,0}, x_{2,0}] + \tilde{x}^* = [0.3107, 1.6893]$. Repeating the formulation and solution of the QP subproblem at iteration 1 produces $[x_{1,2}, x_{2,2}] = [0.5443, 1.9483]$. Note that this solution is infeasible. Continued iterations will produce solutions that converge toward the true solution.



Iteration	$[x_1, x_2]$	$f(x)$	$g(x)$
0	$[1, 1]$	1.3679	0
1	$[0.3107, 1.6893]$	0.8295	-0.4751
2	$[0.5443, 1.9483]$	0.5829	0.0605
3	$[0.5220, 1.9171]$	0.6002	0.0001
4	$[0.5211, 1.9192]$	0.6004	-0.0000

Figure 5 & Table 2: [LEFT] Iso-contours of objective function and constraint for Example 5.1. [RIGHT] Numerical results for first three iterations of SQP. Note that some iterates are infeasible.

SQP provides an algorithmic way to solve NLPs in civil & environmental engineering systems. However, it still relies on approximations - namely truncated Taylor series expansions - to solve the optimization problem via a sequence of QP subproblems. In Chapter 4, we will discuss a direct method for solving NLPs, without approximation.

Exercise 7 (Example 7.10 of [1]). *Consider the NLP*

$$\min_{x_1,x_2} \quad x_1 x_2^{-2} + x_2 x_1^{-1} \tag{47}$$

$$\text{s. to} \quad 1 - x_1 x_2 = 0, \tag{48}$$

$$1 - x_1 - x_2 \leq 0, \tag{49}$$

$$x_1, x_2 \geq 0 \tag{50}$$

with the initial guess $[x_{1,0}, x_{2,0}]^T = [2, 0.5]^T$*. By hand, formulate the Q,R,A,b matrices for the first three iterates. Use Matlab command* `quadprog` *to solve each subproblem. What is the solution after three iterations?*

Exercise 8 (Exercise 7.33 of [1]). *Consider the NLP*

$$\min_{x_1,x_2} \quad 6x_1 x_2^{-1} + x_1^{-2} x_2 \tag{51}$$

$$\text{s. to} \quad x_1 x_2 - 2 = 0, \tag{52}$$

$$1 - x_1 - x_2 \leq 0 \tag{53}$$

with the initial guess $[x_{1,0}, x_{2,0}]^T = [1, 2]^T$*. By hand, formulate the Q,R,A,b matrices for the first three iterates. Use Matlab command* `quadprog` *to solve each subproblem. What is the solution after three iterations?*

6 Notes

Many solvers have been developed specifically for quadratic programs. In Matlab, the `quadprog` command uses algorithms tailored towards the specific structure of QPs.

More information of the theory behind QPs can be found in Section 4.4 of [2], including applications such as minimum variance problems and the famous Markowitz portfolio optimization problem in economics. Readers interested in further details and examples/exercises for optimality conditions should consult Section 4.3 of [1]. Section 7.7 of [1] also provides an excellent exposition of Sequential Quadratic Programming.

Applications of QPs in systems engineering problems include electric vehicles [3], hydro power networks [4], ground-water planning and management [5], transportation engineering [6], financial engineering [7], and structural reliability [8].

References

[1] P. Y. Papalambros and D. J. Wilde, Principles of Optimal Design: Modeling and Computation. Cambridge University Press, 2000.

[2] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2009.

[3] S. Bashash and H. Fathy, “Optimizing demand response of plug-in hybrid electric vehicles using quadratic programming,” in American Control Conference (ACC), 2013, 2013, pp. 716–721.

[4] S. A. A. Moosavian, A. Ghaffari, and A. Salimi, “Sequential quadratic programming and analytic hierarchy process for nonlinear multiobjective optimization of a hydropower network,” Optimal Control Applications and Methods, vol. 31, no. 4, pp. 351–364, 2010. [Online]. Available: http://dx.doi.org/10.1002/oca.909

[5] W. Yeh, “Systems analysis in groundwater planning and management,” Journal of Water Resources Planning and Management, vol. 118, no. 3, pp. 224–237, 1992.

[6] M. E. O’kelly, “A quadratic integer program for the location of interacting hub facilities,” European Journal of Operational Research, vol. 32, no. 3, pp. 393 – 404, 1987.

[7] R. O. Michaud, “The markowitz optimization enigma: Is ’optimized’ optimal?” Financial Analysts Journal, vol. 45, no. 1, pp. pp. 31–42, 1989.

[8] P.-L. Liu and A. D. Kiureghian, “Optimization algorithms for structural reliability,” Structural Safety, vol. 9, no. 3, pp. 161 – 177, 1991.