

ARE211, Fall2015

NPP1: THU, SEP 17, 2015

PRINTED: AUGUST 25, 2015

(LEC# 7)

CONTENTS

3. Nonlinear Programming Problems and the Karush Kuhn Tucker conditions	2
3.1. KKT conditions and the Lagrangian: a “cook-book” example	4
3.1. Separating Hyperplanes	7
3.2. Convex sets, quasi- functions and constrained optimization	8
3.3. The setup	10
3.4. Existence and Uniqueness	11
3.5. Necessary conditions for a solution to an NPP	13
3.6. KKT conditions and the Lagrangian approach	15
3.7. Role of the Constraint Qualification	17
3.8. Binding constraints vs constraints satisfied with equality	20
3.9. Interpretation of the Lagrange Multiplier	22
3.10. Demonstration that KKT conditions are necessary	24
3.11. KKT conditions with equality constraints	29

3. NONLINEAR PROGRAMMING PROBLEMS AND THE KARUSH KUHN TUCKER CONDITIONS

Key points:

- (1) The issues
 - (a) convex sets, quasi- functions and constrained optimization
 - (b) conditions for **existence** of a unique solution
 - (c) **necessary** conditions for \mathbf{x} to be a unique solution
 - (d) **sufficient** conditions for \mathbf{x} to be a unique solution
- (2) The canonical form of the nonlinear programming problem (NPP)

maximize $f(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$,

- (3) Conditions for **existence** of a solution the NPP:

Theorem: If $f : A \rightarrow \mathbb{R}$ is continuous and *strictly quasi-concave*, and A is compact, nonempty and convex, then f attains a unique maximum on A .

- (4) **Necessity** Conditions relating a local maximum on the constraint set to a unique solution to the NPP:

Theorem: If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous and strictly quasi-concave and $g^j : \mathbb{R}^n \rightarrow \mathbb{R}$ is *quasi-convex* for each j , then if f attains a local maximum on $A = \{\mathbf{x} \in \mathbb{R}^n : \forall j, g^j(\mathbf{x}) \leq b_j\}$, this max is the unique global maximum of f on A .

- (5) The KKT conditions in Mantra format:

(Except for a bizarre exception) a necessary condition for \mathbf{x} to solve a constrained maximization problem is that the gradient vector of the objective function at \mathbf{x} belongs to the nonnegative cone defined by the gradient vectors of the constraints that are satisfied with equality at \mathbf{x} .

- (6) The KKT conditions in math format:

If $\bar{\mathbf{x}}$ solves the maximization problem *and the constraint qualification holds* at $\bar{\mathbf{x}}$ then there exists a vector $\bar{\boldsymbol{\lambda}} \in \mathbb{R}_+^m$ such that

$$\nabla f(\bar{\mathbf{x}})^T = \bar{\boldsymbol{\lambda}}^T J\mathbf{g}(\bar{\mathbf{x}})$$

Moreover, $\bar{\boldsymbol{\lambda}}$ has the property that $\bar{\lambda}_j = 0$, for each j such that $g^j(\bar{\mathbf{x}}) < b_j$.

- (7) The distinction between a constraint satisfied with equality and a binding constraint.
- (8) The role of the constraint qualification (CQ):
 - (a) it ensures that the linearized version of the constraint set is, locally, a good approximation of the true nonlinear constraint set.
 - (b) the CQ will be satisfied at \mathbf{x} if the gradients of the constraints that are satisfied with equality at \mathbf{x} form a linear independent set
- (9) Why the KKT conditions are necessary: three special cases.

3.1. KKT conditions and the Lagrangian: a “cook-book” example

The following simple example is included as a stop gap, to give you a template for solving KKT problems, since you need to be able to do this well before we get to the point in the course where you can really understand what’s going on. In the standard introductory problem—maximize utility on a budget set—the budget constraint is virtually always binding. In the following example, it’s not necessarily the case.

Consider the problem of choosing what to eat, when you are on a diet. You have a utility function over three kinds of food—bread, cheese and tomatoes—and you have a calorie constraint. To make this problem look like the standard optimization problem, let p_i denote the number of calories per unit of good i . You’re not allowed to consume more than y calories per day. You have a utility function $u(\mathbf{x})$ over food vectors, and your goal is to solve the following problem

$$\max_{\mathbf{x}} u(\mathbf{x}) \quad \text{s.t.} \quad \sum_{i=1}^n p_i x_i \leq y, \quad \mathbf{x} \geq 0 \quad (1)$$

For this example, we’ll assume the consumer has “Euclidean” preferences, i.e.,

$$u(\mathbf{x}) = -0.5 \sum_{i=1}^n (x_i - \bar{x}_i)^2 \quad (2)$$

That is, she has a “bliss-point” of $\bar{\mathbf{x}}$ which yields her zero utility, and her utility declines quadratically as her consumption bundle moves away from this bliss point. It’s easy to check that

- (1) her indifference curves are concentric circles around her bliss point.
- (2) whether or not her calorie constraint will be binding depends on whether or not the calorie constraint cuts her indifference map to the north east, or south west of her bliss point. In other words, the calorie constraint will be non-binding iff the calorie content of her bliss point, i.e., $\bar{y} = \sum_{i=1}^n p_i \bar{x}_i$, is weakly less than her diet constraint, i.e., iff $\bar{y} \leq y$.

How to proceed:

(1) Set up the Lagrangian:

$$\mathcal{L}(\mathbf{x}, \lambda) = u(\mathbf{x}) + \lambda(y - \sum_{i=1}^n p_i x_i) = - \sum_{i=1}^n 0.5(x_i - \bar{x}_i)^2 + \lambda(y - \sum_{i=1}^n p_i x_i)$$

(2) Differentiate the Lagrangian function w.r.t. each element of \mathbf{x} and also λ .

$$\frac{\partial \mathcal{L}(\mathbf{x}, \lambda)}{\partial x_i} = \frac{\partial u(\mathbf{x})}{\partial x_i} - \lambda p_i = (\bar{x}_i - x_i) - \lambda p_i \leq 0 \quad (3a)$$

$$\frac{\partial \mathcal{L}(\mathbf{x}, \lambda)}{\partial \lambda} = (y - \sum_{i=1}^n p_i x_i) \geq 0 \quad (3b)$$

In addition to these conditions we also have the *complementary slackness conditions*

$$\frac{\partial \mathcal{L}(\mathbf{x}, \lambda)}{\partial x_i} x_i = 0 \quad (3c)$$

$$\frac{\partial \mathcal{L}(\mathbf{x}, \lambda)}{\partial \lambda} \lambda = 0 \quad (3d)$$

At this point, we needn't talk too much about the meaning of the inequalities or the complementary slackness conditions on the x 's. Note simply that the condition $\frac{\partial \mathcal{L}(\mathbf{x}, \lambda)}{\partial \lambda} \geq 0$ says that the dietary constraint must be satisfied: you are allowed to consume less than your allotted calories, but not more.

- (3) First, we'll check if the calorie constraint is going to be nonbinding, i.e., if $\bar{y} \leq y$. If it is then the solution is trivial: $x_i = \bar{x}_i$, for each i . (In general, in order to check whether any of the constraints are binding, you would have to *solve* for the unconstrained (no-diet) solution to the problem. In this simple setting, the solution to the unconstrained problem is, trivially, consume your bliss point.)
- (4) Now assume that the constraint *is* binding, i.e., that $\bar{y} > y$, so that we're going to be on the calorie constraint line. We'll solve the problem for $n = 2$. How does this assumption change our equality/inequality system (3)?
- (a) We replace the inequality in (3b) with an equality
 - (b) We can ignore (3d) because it's now satisfied by assumption.

The next step is to see if we can get a solution to the problem which satisfies the nonnegativity constraints. That is, we will *assume* that there exists a vector \mathbf{x} such that our system

(3a), (3b), and (3c) is satisfied with equalities and see if we reach a contradiction. We'll do it for $n = 2$. (Can also do this for arbitrary n , but it's just more, very tedious algebra.) Specifically, we're going to ignore (3c), replace the inequalities in (3a) with equalities. and solve the three equation system (3a) and (3b) for the three unknowns, x_1 , x_2 and λ .

The cookbook recipe now says

(a) *eliminate λ from the system.* This step is particularly easy when the constraint is linear in the \mathbf{x} vector

(b) *solve a system of two equations in two unknowns.*

Before proceeding, I want to emphasize that for most consumer demand problems, we *can't* do the step I'm going to go through below. The reason is that it's very unusual that you can get an *explicit* solution to the equation system, as we do below. We can do it here because the problem is intentionally simple. But if the utility function were more complicated, e.g., if it were $v(\mathbf{x}) = -\sum_{i=1}^n (x_i - \bar{x}_i)^n/n$, for $n > 3$, we wouldn't ever try to get an explicit solution to the problem. Instead we would invoke the *implicit function theorem* to obtain an *implicit* relationship between endogenous and exogenous variables, and *explicit* expressions for the *derivatives* of these implicit functions w.r.t. exog variables. That is, we wouldn't be able to write down explicitly what the demand function was, but we *would* be able to state explicitly what it's derivatives were. You'll see much, much more of this implicit approach before the year's over. Having said that, back to solving this very special problem explicitly for the demand function.

(a) From the two equations in (3a), we get that $\frac{\bar{x}_1 - x_1}{p_1} = \frac{\bar{x}_2 - x_2}{p_2}$, i.e., that

$$p_1(\bar{x}_2 - x_2) = p_2(\bar{x}_1 - x_1). \quad (4)$$

(b) Letting $\bar{\gamma} = p_1\bar{x}_2 - p_2\bar{x}_1$, we rearrange (4) to get $p_1x_2 - p_2x_1 = \bar{\gamma}$. (Note that $\bar{\gamma}$ could be negative!)

(c) Now, combining this expression with (3b) (replacing " \geq " with " $=$ ") we have two equations in two unknowns

$$p_1x_2 - p_2x_1 = \bar{\gamma} \quad (5a)$$

$$p_2x_2 + p_1x_1 = y \quad (5b)$$

To solve for x_1^* , eliminate x_2 by subtracting $p_2 \times (5a)$ from $p_1 \times (5b)$. This will give us

$$x_1^*(p_2^2 + p_1^2) = p_1y - p_2\bar{\gamma} \implies x_1^* = \frac{p_1y - p_2\bar{\gamma}}{p_2^2 + p_1^2} \quad (5c)$$

To solve for x_2^* , eliminate x_1 by adding $p_1 \times (5a)$ to $p_2 \times (5b)$

$$x_2^*(p_2^2 + p_1^2) = p_2y + p_1\bar{\gamma} \implies x_2^* = \frac{p_2y + p_1\bar{\gamma}}{p_2^2 + p_1^2} \quad (5d)$$

(d) Now, we have to check whether we've reached contradiction. There's no contradiction iff $x_i^* \geq 0$.

(e) Finally, we'll solve for λ^* . From (3b), we have $(\bar{x}_1 - x_1) - \lambda p_1 = 0$, so that

$$\lambda^* = \frac{\bar{x}_1 - x_1^*}{p_1} = \frac{(p_2^2 + p_1^2)\bar{x}_1 - (p_1y - p_2\bar{\gamma})}{p_1(p_2^2 + p_1^2)}$$

3.1. Separating Hyperplanes

A very important property of convex sets is that if they are “almost” disjoint—more precisely, the intersection of their *interiors* is empty—then they can be separated by hyperplanes. For now, we won't be technical about what a hyperplane is. It's enough to know that lines, planes and their higher-dimensional equivalents are all hyperplanes. As Fig. 1 illustrates, this property is completely intuitive in two dimensions.

- In the top two panels, we have convex sets whose interiors do not intersect, and we can put a line between them. Note that it doesn't matter if the sets are disjoint or not. However,
 - if the sets are disjoint, then there will be *many* hyperplanes that separate them.
 - if they are not disjoint, but their interiors do not intersect, then there *may* be a unique hyperplane separating them. What condition guarantees a unique hyperplane? Differentiability of the boundaries of the sets.
- In the bottom two panels, we can't separate the sets by a hyperplane. In the bottom left panel, it's because the sets have common interior points; in the bottom right it's because one set isn't convex.

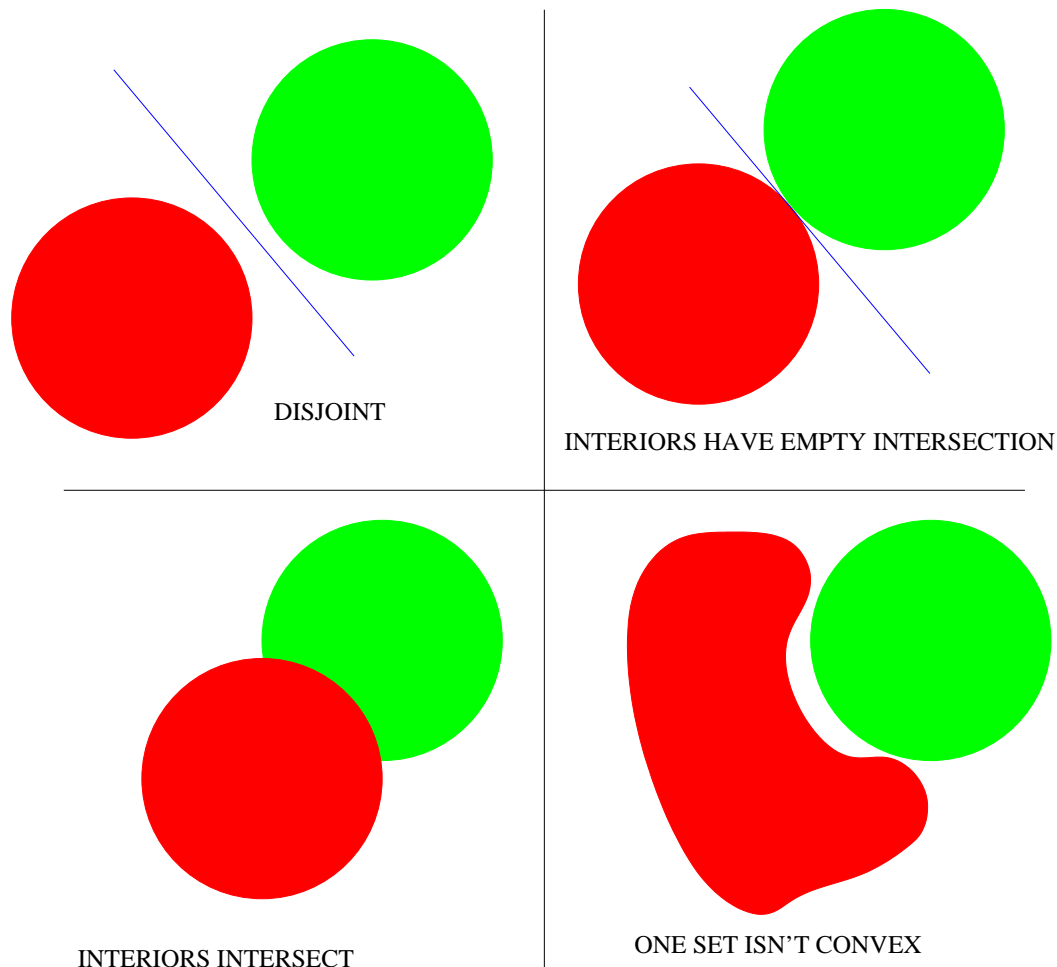


FIGURE 1. Convex sets whose intersection is empty can be separated by a hyperplane

Why do we care about them? They crop up all over the place in economics, econometrics and finance. Two examples will suffice here

- (1) The budget set and the *upper contour set* of a utility function
- (2) The Edgeworth box

Telling you any more would mean teaching you economics, and I don't want to do that!!

3.2. Convex sets, quasi- functions and constrained optimization

Economists want to have a unique solution to the problem of maximizing an objective function subject to a bunch of constraints. For this reason they love convexity.

- Want *upper* contour sets of objective function always to be convex sets
- Want feasible set to be a convex set

Why? Because bad things happen if either isn't true: loosely

- Would like to be able to *separate* objective and upper contour set with a hyperplane.
- Everything *above* the hyperplane is *preferred to* what's on the hyperplane
- All of the feasible stuff is *on or below* the hyperplane
- Implication: optimum on the feasible set has to lie on the hyperplane.
- Analytically, your task then becomes: find the hyperplane.
- unless both upper contour set of objective and feasible set are convex, can't necessarily separate the sets

Role of quasi-ness properties of functions

- A function is *quasi-concave* (*quasi-convex*) if all of its upper (lower) contour sets are convex
- A function is *strictly quasi-concave* (*convex*) if
 - all of its upper (lower) contour sets are “strictly” convex sets (no flat edges)
 - all of its level sets have non-empty interior

The connection between quasi-ness and the constrained optimization problem

- (1) Assume quasi-concave objective function by definition buys you convex upper contour sets
- (2) Define your feasible set as the intersection of lower contour sets of quasi-convex *functions*, e.g., $g^j(\mathbf{x}) \leq b_j$, for $j = 1, \dots, m$.

- each lower contour set is a convex set
- intersection of convex sets is convex

So now you have ensured that your feasible set is convex.

- (3) It's often not natural to describe your feasible set as a bunch of lower contour sets
 - budget set is $\mathbf{p} \cdot \mathbf{x} \leq y; \mathbf{x} \geq 0$.
 - i.e., intersection of a lower contour set with a bunch of *upper* contour sets
 - but
 - *upper* contour sets of $g : \mathbb{R}^n \rightarrow \mathbb{R}$ are always the *lower* contour sets of $(-g)$.

- level sets of $g : \mathbb{R}^n \rightarrow \mathbb{R}$ are always the intersection of a lower and an upper contour set
- so long as all of your constraint functions are quasi-*something*, you can always express your feasible set as the intersection of lower contour sets.

3.3. The setup

The general nonlinear programming problem (NPP) is the following:

$$\text{maximize } f(\mathbf{x}) \text{ subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{b},$$

$$\text{where } f : \mathbb{R}^n \rightarrow \mathbb{R} \text{ and } \mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Terminology:

- f is called the *objective function*;
- \mathbf{g} is a vector of m *constraint functions*, and, of course, $\mathbf{b} \in \mathbb{R}^m$. That is, the individual constraints are stacked together to form a vector-valued function.
- The set of \mathbf{x} such that $\mathbf{g}(\mathbf{x}) \leq \mathbf{b}$ is called the *feasible set* or *constraint set* for the problem.

For the remainder of the course, unless otherwise notified, we will assume that both the objective function and the constraint functions are continuously differentiable. Indeed, we will in fact assume that they are as many times continuously differentiable as we could ever need.

Emphasize that this setup is quite general, i.e., covers every problem you are ever likely to encounter, although for problems involving *equality* constraints you have to modify the conditions.

- can handle constraints of the form $\mathbf{g}(\mathbf{x}) \geq \mathbf{b}$;
- can handle constraints of the form $\mathbf{x} \geq 0$;
- for constraints of the form $\mathbf{g}(\mathbf{x}) \geq \mathbf{b}$, you have to remove the nonnegativity condition on the λ 's.

However,

For example, given $u : \mathbb{R}^2 \rightarrow \mathbb{R}$, consider the problem

$$\text{maximize } u(\mathbf{x}) \text{ subject to } \mathbf{p} \cdot \mathbf{x} \leq y, \mathbf{x} \geq 0;$$

What's g : in this case, g is a linear function, defined as follows:

$$g^0(\mathbf{x}) = \mathbf{p} \cdot \mathbf{x}$$

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

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

so that the problem can be written as

$$\text{maximize } u(\mathbf{x}) \text{ subject to } G\mathbf{x} \leq \mathbf{b},$$

$$\text{where } G = \begin{bmatrix} p_1 & p_2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} y \\ 0 \\ 0 \end{pmatrix}$$

While there are many advantages to having a single, general version of the KKT conditions, the generality comes at a (small) cost. When it comes to actually *computing* the solution to an NPP problem (as opposed to just understanding what's going on), it's very convenient to treat equality constraints *differently* from inequality constraints. This is what matlab does, for example.

3.4. Existence and Uniqueness

The first step is: under what conditions does the NPP have a solution at all? Under what conditions will the solution be unique? We'll answer this question by considering when a function defined on an arbitrary set A attains a maximum and when this maximum is unique. Then we'll apply these results to the NPP, letting A denote the constraint set for the problem. Recall the following theorem, also known as the *extremum value theorem*.

Theorem (Weierstrass): If $f : A \rightarrow \mathbb{R}$ is continuous and A is compact, nonempty, then f attains a maximum on A .

- relationship between A and the constraint sets of the NPP: A is the intersection of the lower contour sets defined by the NPP, i.e., $g_j(\cdot) \leq b_j$, for all j .
- role of compact, i.e., closed and bounded, and nonempty; *not* guaranteed in the specification of the NPP.
 - example of why existence of a maximum may fail if A is not closed: try to maximize $f(x) = 1/x$ on the interval $(0, 1]$.
 - example of why existence of a maximum may fail if A is not bounded try to maximize $f(x) = x$ on \mathbb{R}_+ .
- when you set up the problem, have to make sure the constraint set you end up with is nonempty and compact, otherwise you may not get a solution.

Theorem: If $f : A \rightarrow \mathbb{R}$ is continuous and *strictly quasi-concave*, and A is compact, nonempty and convex, then f attains a unique maximum on A .

- example of why uniqueness may fail if f is not quasi-concave: let A be a circle and construct a function f which has a level set that has two points of tangency with the ball.
- example of why uniqueness may fail if A is not convex: let A be a ball with a bite out of it, and let f be a quasi-concave function whose level set touches the edge of the constraint surface, where the surface has been bitten, at two distinct points.

Theorem: If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous and strictly quasi-concave and $g^j : \mathbb{R}^n \rightarrow \mathbb{R}$ is *quasi-convex* for each j , then if f attains a local maximum on $A = \{\mathbf{x} \in \mathbb{R}^n : \forall j, g^j(\mathbf{x}) \leq b_j\}$, this max is the unique global maximum of f on A .

Proof:

- recall that the *upper* contour sets of a quasi-concave function are convex sets.
- similarly, the *lower* contour sets of a quasi-convex function are convex sets.
- the condition that $g^j(\mathbf{x}) \leq b_j$ is the condition that \mathbf{x} lie in the lower contour set of the j 'th constraint identified by b_j .

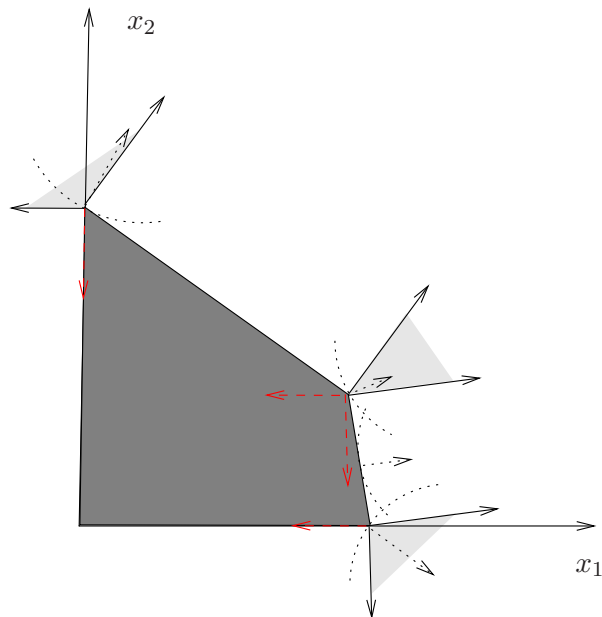


FIGURE 2. The KKT conditions

- finally, the intersection of convex sets is convex.
- now apply previous theorem about uniqueness given conditions on A .

Thus, quasi-convexity of g^j 's guarantees convexity of A . However, it doesn't imply COMPACTNESS, hence existence isn't guaranteed.

Take stock: haven't told you how to solve one of these problems, only when it will have a solution and when that solution will be unique.

3.5. Necessary conditions for a solution to an NPP

In this topic, we'll give two theorems, one is necessary and one is sufficient for $\bar{\mathbf{x}}$ to be a maximum. For now, we're focusing on necessary conditions. First, the theorem in words, in Mantra format:

Theorem (Kuhn Tucker necessary conditions) as a Mantra: (Except for a bizarre exception) a necessary condition for \mathbf{x} to solve a constrained maximization problem is that the gradient vector of the objective function at \mathbf{x} belongs to the nonnegative cone defined by the gradient vectors of the constraints that are satisfied with equality at \mathbf{x} .

The bizarre exception is called the *constraint qualification*. We'll get to it in a minute.

Now for the formal statement of the theorem.

Theorem (Kuhn Tucker necessary conditions):¹ If $\bar{\mathbf{x}}$ solves the maximization problem *and the constraint qualification holds* at $\bar{\mathbf{x}}$ then there exists a vector $\bar{\boldsymbol{\lambda}} \in \mathbb{R}_+^m$ such that

$$\nabla f(\bar{\mathbf{x}})^T = \bar{\boldsymbol{\lambda}}^T J\mathbf{g}(\bar{\mathbf{x}}) \quad (6)$$

Moreover, $\bar{\boldsymbol{\lambda}}$ has the property that $\bar{\lambda}_j = 0$, for each j such that $g^j(\bar{\mathbf{x}}) < b_j$.

The scalar λ_j is known as the *lagrange multiplier* corresponding to the j 'th constraint. Note that $\boldsymbol{\lambda}^T$ denotes the transpose of $\boldsymbol{\lambda}$, i.e., if $\boldsymbol{\lambda}$ is assumed to be a column vector, then $\boldsymbol{\lambda}^T$ will be a row vector. Why didn't I write $J\mathbf{g}(\bar{\mathbf{x}})\boldsymbol{\lambda}$?

- Recall that if $\mathbf{b} = A\mathbf{w}$, then \mathbf{b} is a weighted linear combination of the *columns* of A .
- Imagine if you wrote $\mathbf{b} = \mathbf{w}'A$, then what would \mathbf{b} be? Ans.: it would be a weighted linear combination of the *rows* of A . (In our case
 - \mathbf{b} is $\nabla f(\mathbf{x})$
 - A is a matrix, each *row* of which is the gradient of one constraint
 - \mathbf{w} corresponds to the vector of weights $\boldsymbol{\lambda}$.)
- That's what we want here: in words, the theorem says: if $\bar{\mathbf{x}}$ solves the maximization problem *and the constraint qualification holds* then the gradient of f at $\bar{\mathbf{x}}$ is a nonnegative weighted linear combination of the gradients of the constraints that are satisfied with equality.

¹ Actually, this is not the Kuhn Tucker theorem but the Karush Kuhn Tucker theorem. Karusch was a graduate student who was partly responsible for the result. Somehow, his name has been forgotten. A regrettable thing for graduate students

3.6. KKT conditions and the Lagrangian approach

The version of the KKT conditions using the mantra is intuitive and lends itself to diagrammatic explanations, etc. For some people, it's a bit abstract however. To actually solve Lagrangian problems, most people typically proceed by setting up a *Lagrangian* objective, for which they compute first order conditions. What's important to understand is that the Lagrangian route is a long way around to end up at *exactly* the same place that we've gotten to with the Theorem on p. 14; the FOC for the Lagrangian is nothing other than a line-by-line representation of the KKT necessary conditions that display (6) expresses in vector (or matrix) form.

The Lagrangian function for the canonical NPP is as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}(\mathbf{b} - \mathbf{g}(\mathbf{x})) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j (b_j - g^j(\mathbf{x}))$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^m$. Note that $\boldsymbol{\lambda}$ is treated just like \mathbf{x} , i.e., as a variable to be optimized. The first order conditions for an extremum of \mathcal{L} on the space $\mathbb{R}^n \times \mathbb{R}_+^m$ are:

$$\text{for } i = 1, \dots, n, \quad \frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial x_i} = 0; \quad (7a)$$

$$\text{for } j = 1, \dots, m, \quad \frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial \lambda_j} \geq 0 \quad (7b)$$

$$\text{for } j = 1, \dots, m, \quad \bar{\lambda}_j \frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial \lambda_j} = 0. \quad (7c)$$

Note well: this specification of the Lagrangian first order conditions differs from many textbooks, eg., Chiang, which write the FOC w.r.t. \mathbf{x} as $\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}) / \partial x_i \leq 0$. The reason for the difference is that these textbooks are adding an additional restriction on their specification of the NPP, i.e., that $\mathbf{x} \geq 0$. As you recall, my specification of the canonical problem treats the requirement that $\mathbf{x} \geq 0$ as just n more inequality constraints, i.e., $\mathbf{x} \geq 0$.

You'll observe that I didn't say FOC for a maximum of \mathcal{L} ; technically, what we're looking for is a maximum of \mathcal{L} w.r.t. \mathbf{x} and a *minimum* of \mathcal{L} w.r.t. $\boldsymbol{\lambda}$, subject to the additional constraint that the $\boldsymbol{\lambda}$ vector is nonnegative. It would be highly appropriate to ask, at this point, *why on earth*

would anybody ever consider looking for such a weird object as a max w.r.t. \mathbf{x} and a min w.r.t. $\boldsymbol{\lambda}$ s.t. $\boldsymbol{\lambda} \geq 0$? It's not highly intuitive. In fact, everybody always just talks about "the first order conditions for the Lagrangian" and nobody bothers to wonder what it means, except in classes like this. I strongly recommend not bothering to ask this highly appropriate question; instead, think of the Lagrangian is nothing but a construction that delivers us the KKT conditions, i.e.,

Theorem: A vector $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$ satisfies the KKT conditions (6) iff it satisfies (7)

To prove the theorem, recall the part of the KKT condition which states

$$\nabla f(\bar{\mathbf{x}})^T = \bar{\boldsymbol{\lambda}}^T J\mathbf{g}(\bar{\mathbf{x}}) \quad (6)$$

On the other hand, writing out $\frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial x_i} = 0$, we have

$$\frac{\partial f(\bar{\mathbf{x}})}{\partial x_i} = \sum_{j=1}^m \bar{\lambda}_j \cdot \frac{\partial g^j(\bar{\mathbf{x}})}{\partial x_i} \quad (8)$$

The left hand side of (8) is the i 'th element of the vector $\nabla f(\bar{\mathbf{x}})$. The right hand side is the inner product of $\bar{\boldsymbol{\lambda}}$ with the i 'th row of the matrix $J\mathbf{g}(\bar{\mathbf{x}})$. Thus, the first order conditions for the x_i 's are simply a long-hand way of writing out what (6) states concisely in vector notation. Now consider the m conditions on the λ_j 's, i.e., $\frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial \lambda_j} \geq 0$, or $b_j \geq g^j(\bar{\mathbf{x}})$. Taken together these m conditions state that $\bar{\mathbf{x}}$ belongs to the intersection of the lower contour sets of the g^j 's corresponding to the b_j 's. Finally, consider the condition $\bar{\lambda}_j \frac{\partial \mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})}{\partial \lambda_j} = 0$, which is known as a *complementary slackness* condition. This condition will be satisfied iff *either* $\bar{\lambda}_j = 0$. *or* $b_j = g^j(\bar{\mathbf{x}})$ *or* both equalities are satisfied simultaneously.

An important property of the Lagrangian specification is that the value of the Lagrangian function at the solution to the NPP is equal to the maximized value of the objective function on the constraint set. In symbols:

Theorem: At a solution $(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}})$ to the NPP, $\mathcal{L}(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}) = f(\bar{\mathbf{x}})$.

To see that this is correct note that $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}(\mathbf{b} - \mathbf{g}(\mathbf{x}))$. Each component of the inner product $\boldsymbol{\lambda}(\mathbf{b} - \mathbf{g}(\mathbf{x}))$ is zero, since *either* $(b_j - g^j(\mathbf{x}))$ is zero *or* λ_j is zero. Hence the entire second term on the right hand side of the Lagrangian is zero.

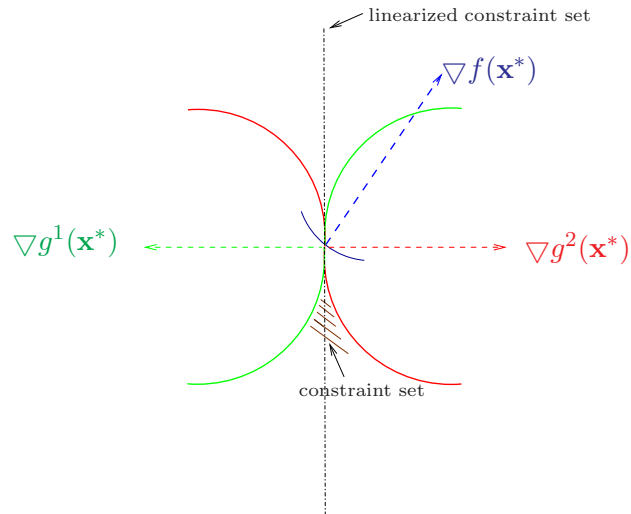


FIGURE 3. Varian's "Hershey Kiss" example: the Constraint Qualification is violated

3.7. Role of the Constraint Qualification

We'll now show that in the absence of the caveat about the constraint qualification, the KKT theorem above would be false.

- the constraint qualification relates to the mysterious "bizarre exception" that is added as a caveat to the mantra.
- In Fig. 3 below, the gradient vectors for the constraints satisfied with equality are not linearly independent, and don't span the domain of the objective function, so that you can't write the objective function as a linear combination of them. This doesn't mean that we don't have a maximum at the tip of the constraint set.
- A sufficient condition for the CQ to be satisfied is that the gradients of the constraints satisfied with equality at $\bar{\mathbf{x}}$ form a linearly independent set.
- Another way of stating the above is as follows: let $M(\mathbf{x})$ denote what's left of $Jg(\mathbf{x})$ (i.e., the Jacobian matrix of \mathbf{g} evaluated at \mathbf{x}), *after you've eliminated all rows of this matrix corresponding to constraints satisfied with strict inequality at \mathbf{x}* ; a sufficient condition for the CQ to be satisfied at \mathbf{x} is that the matrix $M(\mathbf{x})$ has full rank.
- If the constraint qualification isn't satisfied then you could have a maximum at $\bar{\mathbf{x}}$ but wouldn't know about it by looking at the KKT conditions. at this point.

- The right way to think about what goes wrong is that when the CQ is violated, the information conveyed by the gradient vectors doesn't accurately describe the constraint set.
- Saying this more precisely, when we use the KKT conditions to solve an NPP, what we are actually doing is solving the *linearized version* of the problem. By this I mean, we replace the boundary of the constraint set, as defined by the level sets of the constraint functions, by the piecewise linear surface defined by the *tangent planes* to the level sets. In order for the KKT conditions to do their job, it had better be the case that in a neighborhood of the solution you are considering, this "linear shape" looks, *locally*, pretty much like the shape of the true nonlinear constraint set for the problem.
- In the case of our example, the tangent lines to the two level sets, at the tip of the pointy cone, are parallel to each other, so that the "linearized problem" is a line that goes on for ever.
 - Given the objective function that we've drawn, the linearized problem has *no* solution, you just go on and on forever along the line.
 - the constraint set for the true problem doesn't look anything like a line, so that the linearized problem is completely misleading
 - observe that if you "bumped" the original problem a little bit, so that the tangent planes at the tip were not parallel to each other, then the close relationship between the original and the linearized problems would be restored.
 - what determines whether the linearized problem accurately represents the original problem? Answer is apparent from comparing the relationship between the gradient vectors to the tangent planes for the original, pathological problem to this relationship for the bumped, well-behaved problem: in the former case, the two gradient vectors are collinear.
 - More generally, the linearized problem will *locally* accurately represent the original nonlinear problem at a point if the gradient vectors of the constraints that are satisfied with equality at that point form a linear independent set.
- You'll notice that neither of the constraint functions are quasi-convex, and that the constraint set is not convex. A natural thing to hope is that imposing quasi-convexity of the

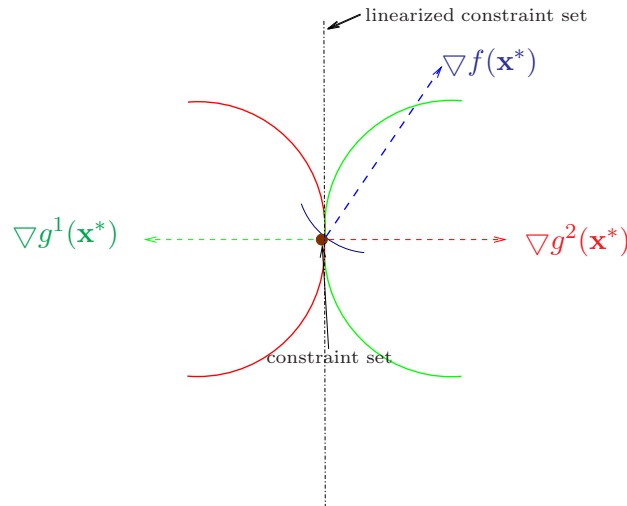


FIGURE 4. An NPP with quasi-convex constraints which fails the CQ

constraint functions (convexity of the constraint set) would make this problem go away. It doesn't, as Fig. 4 illustrates: it's a silly NPP problem because the constraint set is a singleton. With some fiddling, however, we could easily fix that and make the constraint set bigger.

It follows from the above that there are two special cases in which the constraint qualification is satisfied vacuously.

- (1) if all of the constraints are linear, (more precisely, if all of the constraint functions have affine level sets). In this case, trivially, the linearized problem will be identical to the original problem.
- (2) if there is only one constraint, whose gradient is non-zero

It's important to keep in mind that the constraint qualification is a *sufficient* condition that ensures that the KKT conditions are *necessary* for a solution. Some people have wondered the following: "could I find a point that satisfies the KKT conditions, leading me to believe that I've found a solution to the NPP, but then the constraint qualification turns out to be violated, so that candidate solution turns out not to be a solution?" The answer to this question is a resounding NO. If the constraint qualification is violated, there may be solutions to the NPP that *don't* satisfy the KKT conditions.

If you say that finding a point that satisfies the KKT conditions is a “positive” and not finding such a point is a “negative”, then we can talk about “false positives” (in the sense of a positive result of a test, even though the thing you’re testing for (say pregnancy) hasn’t happened) and “false negatives” (the test says you aren’t pregnant when you are).

In KKT theory, we’ve seen both kinds of “falseness”: there are several ways that we can get “false positives” in the sense that the KKT conditions are satisfied at some point which *does not* solve the NPP. We’ll see some of these in a minute. However, provided your function is differentiable and the constraint set is compact, the only way to get a “false negative” is for the CQ to be violated (e.g., Varian’s “Hershey Kiss” example (Fig. 3 below)). Thus, provided the CQ is not violated, there will be no “false negatives” and the maximum (if it exists) must satisfy KKT.

3.8. Binding constraints vs constraints satisfied with equality

In some math-for-economists text-books (e.g., Simon and Blume), the term “constraint satisfied with equality” is (unfortunately) replaced by “binding constraints”. This is unfortunate because in practice, economists almost always reserve the term “binding” for a subclass of the constraints satisfied with equality. Specifically, a constraint is said to be *binding* if the maximized value of the objective function increases when the constraint is relaxed a little bit. (A constraint is relaxed by increasing the constant on the right hand side of the constraint inequality.) For example in Fig. 5, both constraints are satisfied with equality, but only g^1 is binding; if you relax constraint g^1 (left panel of Fig. 6) the optimum moves to the north-east; if you relax constraint g^2 (right panel of Fig. 6) , the solution is unchanged.

Formally, consider the problem

$$\max f(\mathbf{x}) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{b}$$

Let $\mu(\mathbf{b})$ denote the solution to the problem with constraint vector \mathbf{b} . We now say that constraint j is *binding* if there exists $\bar{\epsilon} > 0$ such that $\forall \epsilon \in (0, \bar{\epsilon}), \mu(\mathbf{b} + \epsilon e^j) > \mu(\mathbf{b})$, where e^j denotes the j 'th unit vector. In this course we’ll always say that a constraint is “satisfied with equality” unless we know that it’s actually binding in the sense above.

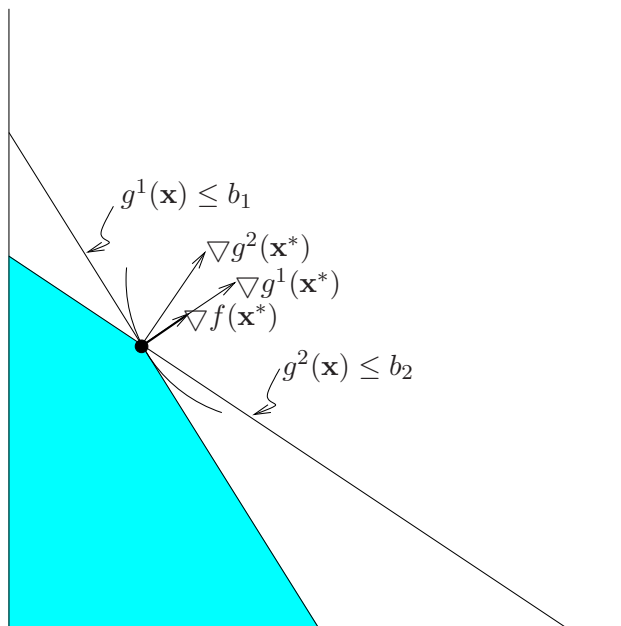


FIGURE 5. constraint g^1 is binding, but g^2 isn't

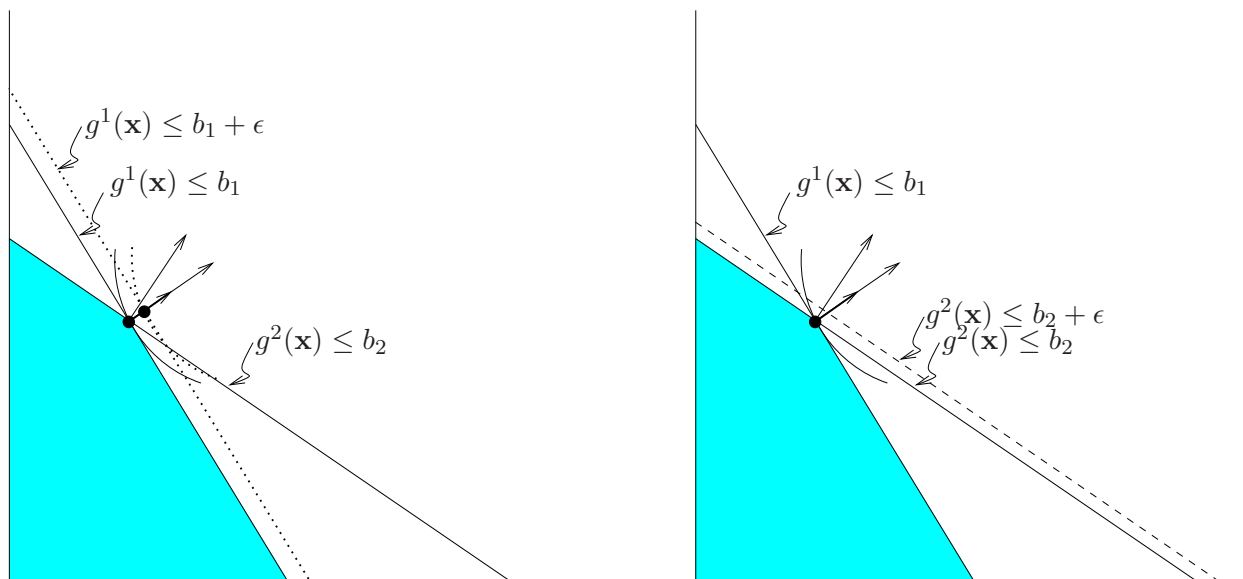


FIGURE 6. Relaxing the two constraints

Economists almost always decide whether a constraint is binding or merely satisfied with equality depending on whether the corresponding lagrange multiplier is positive or zero. This is almost but not quite correct: positivity of the lagrangian is *sufficient* but not necessary for bindingness. For example, consider $\max f(x) = x^3$, such that $g(x) = x \leq b = 0$. The solution to this problem is $x^* = 0$. At this point, $f'(x^*) = 0 \times g'(x^*)$, i.e., the lagrangian is zero. But the constraint is

certainly binding. We've seen this distinction over and over again in this course. What's going on here is exactly the same as what's going on when we say that negative definiteness of the Hessian is sufficient but not necessary for strict concavity.

3.9. Interpretation of the Lagrange Multiplier

The Lagrangian λ_j has an interpretation that proves to be important in a wide variety of economic applications.

- it is a measure of how much bang for the buck you get when you relax the j 'th constraint. That is, if you *increase* b_j by one unit, then the maximized value of your objective function will go up by λ_j units.
- Example: think of maximizing utility on a budget constraint. the higher are prices the longer is the gradient vector, and so the shorter is λ , i.e., the less bang for the buck you get, literally. I.e., relaxing your budget constraint by a dollar doesn't buy you much more of a bundle, so that your utility can't go up by much. On the other hand, for a fixed price vector, the length of the gradient of your utility function is a measure of how easy to please you are. If you are a kid who gets a lot of pleasure out of penny candy, then relaxing the budget constraint by a dollar will buy you a lot.
- the importance of this in economic applications is that you often want to know what the economic cost of a constraint is, e.g., suppose you are maximizing output subject to a resource constraint: what's it worth to you to increase the level of available resources by one unit. Get answer by looking at the Lagrangian.
- The mathematical proof is as usual completely trivial. Let $M(\mathbf{b})$ denote the maximized value of the objective function when the constraint vector is \mathbf{b} .

$$M(\mathbf{b}) = f(\bar{\mathbf{x}}(\mathbf{b})) = \mathcal{L}(\bar{\mathbf{x}}(\mathbf{b}), \bar{\boldsymbol{\lambda}}(\mathbf{b})) = f(\mathbf{x}) + \boldsymbol{\lambda}(\mathbf{b} - \mathbf{g}(\mathbf{x})).$$

we have

$$\begin{aligned} \frac{dM(\mathbf{b})}{db_j} = \frac{d\mathcal{L}(\bar{\mathbf{x}}(\mathbf{b}), \bar{\lambda}(\mathbf{b}))}{db_j} &= \sum_{i=1}^n \left\{ f_i(\bar{\mathbf{x}}(\mathbf{b})) - \sum_{k=1}^m \lambda_k g_i^k(\bar{\mathbf{x}}(\mathbf{b})) \right\} \frac{dx_i}{db_j} \\ &+ \sum_{k=1}^m \frac{d\lambda_k}{db_j} (b_k - g^k(\mathbf{x})) + \lambda_j \end{aligned}$$

- For each i , the term in curly brackets is zero, by the KKT conditions.
- For each k ,
 - * If $(b_k - g^k(\mathbf{x})) = 0$, then $\frac{d\lambda_k}{db_j}(b_k - g^k(\mathbf{x}))$ is zero.
 - * If $(b_k - g^k(\mathbf{x})) < 0$, then $\lambda_k(\cdot)$ will be zero on a neighborhood of \mathbf{b} , so that $\frac{d\lambda_k(\cdot)}{db_j} = 0$.
- The only term remaining is λ_j .
- Conclude that $\frac{dM(\mathbf{b})}{db_j} = \lambda_j$
- Here's a more intuitive approach for the case of a single binding constraint.
 - increase the constraint b to $b + db$.
 - you should respond by moving from solution \mathbf{x} in the direction of steepest ascent, i.e., in the direction that the gradient is pointing.
 - move from \mathbf{x} in the direction that $\nabla f(\mathbf{x})$ is pointing until you reach the new constraint, i.e., define $d\mathbf{x}$ by $g(\mathbf{x} + d\mathbf{x}) = b + db$. Now initially we had $g(\mathbf{x}) = b$. Hence

$$db = g(\mathbf{x} + d\mathbf{x}) - g(\mathbf{x}) \approx \nabla g(\mathbf{x})d\mathbf{x}$$

while

$$df = \nabla f(\mathbf{x})d\mathbf{x}$$

which by the KKT conditions

$$= \lambda \nabla g(\mathbf{x})d\mathbf{x}$$

Hence

$$df = \lambda db$$

Hence (being a little fast and loose with infinitesimals), $df/db = \lambda$.

Note that λ will be larger

- the more rapidly f increases with \mathbf{x} (i.e., the longer is the vector $\nabla f(\mathbf{x})$).
- the *less* rapidly g increases with \mathbf{x} (i.e., the *shorter* is the vector $\nabla g(\mathbf{x})$).

Note also that the above “proves” the 2nd part of the mantra, i.e., if constraint j is satisfied with equality but is not binding, then the weight on this constraint must be zero when you write $\nabla f(\mathbf{x})$ as a nonnegative linear combination of the $\nabla g^j(\mathbf{x})$ ’s. The following argument proves this rather sloppily: (a) if constraint j is satisfied with equality but is not binding, then by definition $\frac{df}{db_j} = 0$; (b) since $\frac{df}{db_j} = \lambda_j$, then λ_j must be zero.

Fig. 7 illustrates what’s going on. In the left panel,

- $\|\nabla f\|$ is small and $\|\nabla g\|$ is big;
- since $\|\nabla g\|$ is big, only a *small* \mathbf{dx} is required in order to increase g by one unit.
- since $\|\nabla f\|$ is small, this small \mathbf{dx} increases f by only a little bit.

In the right panel, everything is reversed:

- $\|\nabla g\|$ is small and $\|\nabla f\|$ is big;.
- Since $\|\nabla g\|$ is small, a *big* \mathbf{dx} is required in order to increase g by one unit.
- Since $\|\nabla f\|$ is big, this big \mathbf{dx} increases f by a lot.

3.10. Demonstration that KKT conditions are necessary

Rather than prove the result generally, we are just going to look at special cases. Throughout this subsection, we are going to ignore the question of whether the KTT conditions are sufficient for a solution. The pictures we are going to consider are of functions so well behaved that if the KTT conditions are satisfied at a point then that point is a solution.

The case of one inequality constraint: With one constraint, the picture is easy: the condition is

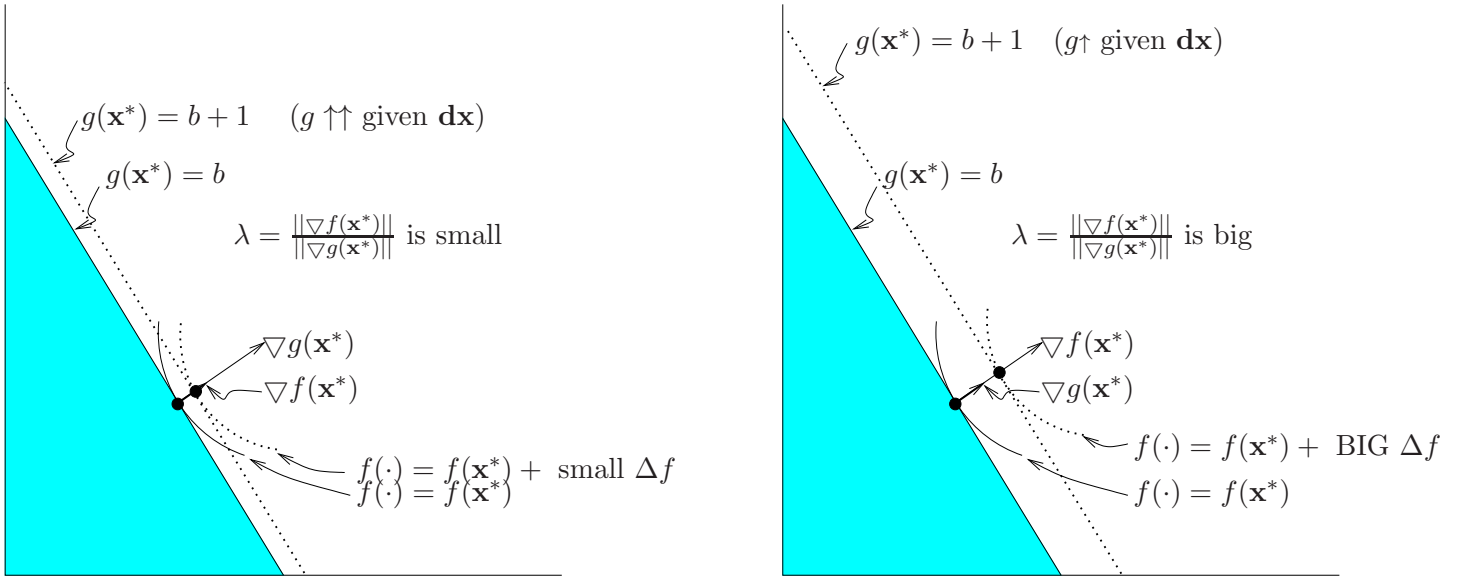


FIGURE 7. Interpretation of the Lagrangian

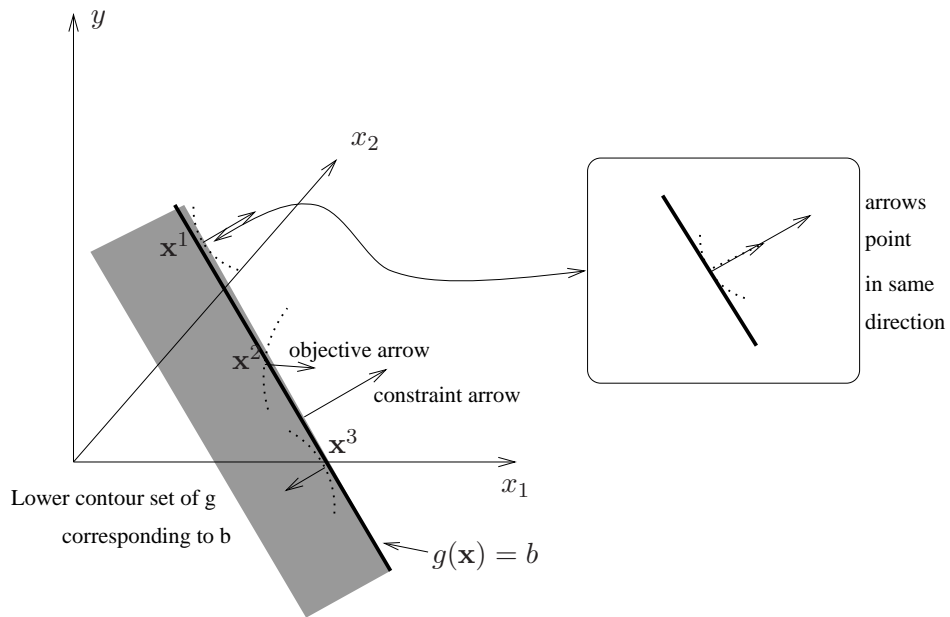


FIGURE 8. Constrained max problem with one inequality constraint

that $\nabla f(\bar{\mathbf{x}}) = \lambda \nabla g(\bar{\mathbf{x}})$, for some nonnegative scalar λ . Also, with one constraint the CQ is vacuous. In Fig. 8

- The constraint is the condition that $g(\mathbf{x}) \leq b$. That is, the only \mathbf{x} 's we are allowed to consider are \mathbf{x} 's for which the function g assigns values weakly less than b .
- Three candidates in the picture \mathbf{x}^1 , \mathbf{x}^2 and \mathbf{x}^3 . Each curve represents a level set of some function thru one of these points. Note: each level set belongs to a *different* function!
- Point \mathbf{x}^2 isn't a maximum; upper contour set intersects with constraint set; Point \mathbf{x}^3 isn't either. Point \mathbf{x}^1 is the only one with the property that there's nothing in the *strict* upper contour set that is also in the constraint set.
- Now put in the arrows. Call the first the *objective arrow* (i.e., the gradient of the objective function) and the second the *constraint arrow* (i.e., the gradient of the constraint function). Which way do the objective arrows point?
 - arrow points in direction of increase
 - this must be the upper contour set
 - upper contour set is convex
 - Conclude: we know that the arrow for a level set points into the corresponding *upper contour* set, which for a quasi-concave function is the *convex* set bounded by the set.
- Which way does the constraint arrow point? Must be NE because the lower contour set is SW.
- Now characterize the condition for a maximum in terms of the arrows. Answer is that arrows lie on top of each other and point in the same direction (as opposed to lying on top of each other but point at 180 degrees to each other).
- Conclude in terms of the mantra: arrow lies in the positive cone defined by the unique binding constraint.

Question: Suppose that for each of the level sets of f , the gradient vector at every point on this level set points SW as in point \mathbf{x}^3 ? What can you conclude about the solution to the maximization problem?

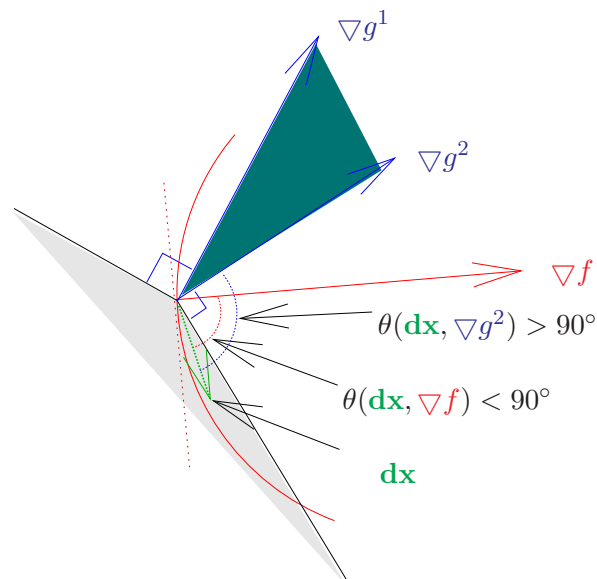


FIGURE 9. Graphical Illustration of the KKT conditions: two constraints

Answer: There is no solution. Certainly there can't be a solution in which the constraint is binding. And there can't be an unconstrained solution either. You can *always* move a bit further southwest and increase the value of f

More formally, for the case of one inequality constraint, the KKT condition is that there is a *nonnegative collinear* relationship between the gradients of the objective and the constraint functions. To see that this condition is necessary, suppose you don't have nonnegative collinearity, i.e., $\nabla f(\bar{\mathbf{x}})$ is not a nonnegative scalar multiple of $\nabla g(\bar{\mathbf{x}})$. Then you can find a vector \mathbf{dx} such that $\nabla g(\bar{\mathbf{x}}) \cdot \mathbf{dx} < 0$ and $\nabla f(\bar{\mathbf{x}}) \cdot \mathbf{dx} > 0$. (Takes a bit of linear algebra work to prove this but it's obvious diagrammatically.) But this means that $\mathbf{x} + \mathbf{dx}$ satisfies the inequality constraint and (by Taylor's theorem) increases f provided \mathbf{dx} is sufficiently small.

The two constraint case: With two constraints the condition says that at the solution the gradient vector $\nabla f(\bar{\mathbf{x}})$, can be written as a nonnegative linear combination of the gradients of the constraints that are satisfied with equality

What does this mean geometrically:

- means that the gradient vector of the objective function points into the cone defined by the gradients of the constraints that are satisfied with equality. Go over what it means for a vector to be inside the positive cone defined by two others. Show geometrically that if a vector \mathbf{x} is inside the cone, you can take positive scalar multiples of the vectors that define the cone, and reconstruct \mathbf{x} . Otherwise, you can't reconstruct \mathbf{x} with positive coefficients.
- to see why must this condition hold geometrically, we draw a picture (Fig. 9) that has *only* the gradients drawn in (none of the constraint sets or indifference curves will be drawn in) and argue from the vectors *alone* why the above condition is necessary.
- Suppose the above condition is violated, so that $\nabla f(\bar{\mathbf{x}})$ lies outside the cone defined by the gradients of the constraints that are satisfied with equality. Clearly, we can then draw a line (the horizontal dotted line in Fig. 9) such that the gradient vectors of all of the constraints lie on one side of the line, and the gradient vector of the objective function lies on the other side. (Again this is obvious geometrically, but requires some work to prove rigorously.)
- We can now draw a vector \mathbf{dx} that makes an acute angle with $\nabla f(\bar{\mathbf{x}})$ but an obtuse angle with all of the constraint vectors. We can *always* do this: make \mathbf{dx} arbitrarily close to the line perpendicular to the first dotted line, which ensures that \mathbf{dx} will make an obtuse angle with the constraint gradients, but must make an acute angle with $\nabla f(\bar{\mathbf{x}})$ since both vectors are trapped within the quadrant defined by the dotted lines.
- Observe that it makes an obtuse angle with all of the constraint vectors.
- Reason from there:
 - add \mathbf{dx} to $\bar{\mathbf{x}}$
 - observe that $f(\bar{\mathbf{x}} + \mathbf{dx}) - f(\bar{\mathbf{x}}) \approx \nabla f(\bar{\mathbf{x}}) \cdot \mathbf{dx} > 0$;
 - i.e., moving in this direction increases the objective function.
 - similarly, for all j such that the j 'th constraint is satisfied with equality, observe that $g^j(\bar{\mathbf{x}} + \mathbf{dx}) - g^j(\bar{\mathbf{x}}) \approx \nabla g^j(\bar{\mathbf{x}}) \cdot \mathbf{dx} < 0$, i.e., reduces the value of all of the constraints that are satisfied with equality.
 - for j 's that aren't, if \mathbf{dx} is sufficiently small, then you can increase the value of $g^j(\cdot)$ a little bit and still be less than b^j .

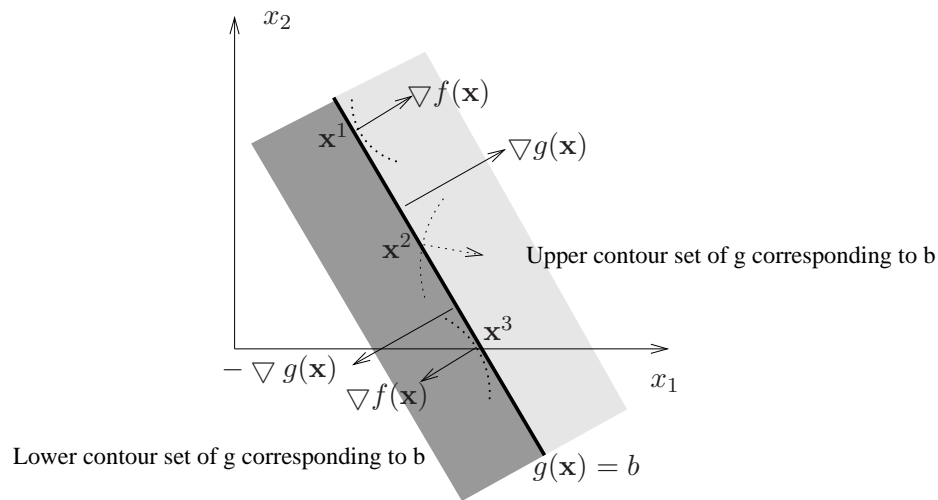


FIGURE 10. Maximization subject to an equality constraint

- in other words, you can move a little in the direction of \mathbf{dx} and stay within the constraint set, yet increase the value of f .
- This establishes that $\bar{\mathbf{x}}$ couldn't have been a maximum.

3.11. KKT conditions with equality constraints

One equality constraint, $g(\mathbf{x}) = b$: Consider the problem

$$\max f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) = b;$$

- This is a much more restrictive problem than the corresponding problem with inequality constraints:
 - in this problem the solution has to belong to the *level* set of $g(\cdot)$ corresponding to b .
 - if the constraint were an inequality, \mathbf{x} could be anywhere in the *lower contour* set of $g(\cdot)$ corresponding to b .
- as the picture illustrates
 - it's still the case that a necessary condition for \mathbf{x} to solve the problem is that $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are colinear

- but in this problem, it's possible for $\nabla g(\mathbf{x})$ to be a *negative* scalar multiple of $\nabla f(\mathbf{x})$.
 - * in this case, the $\nabla f(\mathbf{x})$ would point into the lower contour set of g corresponding to b .
- in other words, the nonnegativity constraint on λ no longer applies.
- the necessary condition is now

$$\nabla f(\bar{\mathbf{x}}) = \lambda \nabla g(\bar{\mathbf{x}}), \quad \lambda \in \mathbb{R}$$

In terms of the mantra, we replace the term “nonnegative cone” with “subspace”.

Multiple equality constraints: $\mathbf{g}(\mathbf{x}) = \mathbf{b}$:

The KKT conditions for equality constrained problems: Mantra format:

(Except for a bizarre exception) a necessary condition for \mathbf{x} to solve a constrained maximization problem in which all constraints must hold with equality is that the gradient vector of the objective function at \mathbf{x} belongs to the subspace defined by the gradient vectors of the constraints that are satisfied with equality at \mathbf{x} .

The KKT conditions for pure equality constrained problems: math format:

maximize $f(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) = \mathbf{b}$,

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$

If $\bar{\mathbf{x}}$ solves the maximization problem *and the constraint qualification holds* at $\bar{\mathbf{x}}$ then there exists a vector $\bar{\boldsymbol{\lambda}} \in \mathbb{R}^m$ such that

$$\nabla f(\bar{\mathbf{x}})^T = \bar{\boldsymbol{\lambda}}^T J\mathbf{g}(\bar{\mathbf{x}})$$

Equality and Inequality constraints

The KKT conditions for problems with equality and inequality constraints: math format:

maximize $f(\mathbf{x})$ subject to $\mathbf{g}^1(\mathbf{x}) \leq \mathbf{b}^1, \mathbf{g}^2(\mathbf{x}) = \mathbf{b}^2$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{g}^i : \mathbb{R}^n \rightarrow \mathbb{R}^{m^i}$

If $\bar{\mathbf{x}}$ solves the maximization problem *and the constraint qualification holds* at $\bar{\mathbf{x}}$ then there exists vectors $\bar{\boldsymbol{\lambda}}^1 \in \mathbb{R}_+^{m^1}$, $\bar{\boldsymbol{\lambda}}^2 \in \mathbb{R}^{m^2}$ such that

$$\nabla f(\bar{\mathbf{x}}) = \bar{\boldsymbol{\lambda}}^1 J\mathbf{g}^1(\bar{\mathbf{x}}) + \bar{\boldsymbol{\lambda}}^2 J\mathbf{g}^2(\bar{\mathbf{x}})$$

Moreover, $\bar{\boldsymbol{\lambda}}^1$ has the property that $\bar{\lambda}_j^1 = 0$, for each j such that $g_j^1(\bar{\mathbf{x}}) < b_j^1$.