appendix one

Constrained optimization

The aim of this appendix is to give you a bare working knowledge of how to solve problems of the form

\[
\text{maximize } f(x) \\
\text{subject to } g_i(x) \leq c_i \text{ for } i = 1, \ldots, n
\]

for functions \( f, g_1, g_2, \ldots, g_n \) whose domain is \( \mathbb{R}^k \) for some integer \( k \) and whose range is \( \mathbb{R} \), and for constants \( c_1, c_2, \ldots, c_n \). The mathematical theory of optimization with inequality constraints concerns all the things discussed here, and this discussion won't get you within shouting distance of that theory. So while this appendix will tell you enough to make it through the applications in the book, you should engage in further rigorous study of this material; two recommendations are made at the end of the appendix.

We'll proceed as follows. First, in cookbook fashion, the recipe for finding solutions to this sort of problem will be given. Then an example will be worked to illustrate the recipe. And then, by looking at the simplest possible example we'll try to give some intuition for the recipe.

**A1.1. A recipe for solving problems**

We want to solve

\[
\text{maximize } f(x) \\
\text{subject to } g_i(x) \leq c_i \text{ for } i = 1, \ldots, n.
\]

Here, mechanically, is what to do.

*Step 1. Form the Lagrangian.*
Appendix one: Constrained optimization

For each of the \( n \) constraints, create a multiplier — this is a real variable. The multiplier for the constraint \( g_i(x) \leq c_i \) will be denoted by \( \lambda_i \). Then the Lagrangian is the function

\[
f(x) = \sum_{i=1}^{n} \lambda_i g_i(x).
\]

(In some books the term for the \( i \)th constraint is \( \lambda_i(g_i(x) - c_i) \). It will make no difference for the form of the recipe that I'm giving you here.) This should be thought of as a function of \( k + n \) variables, namely the \( k \) components of \( x \) and the \( n \) \( \lambda \)s. We'll write \( L(x, \lambda) \) for this function, where \( \lambda \) should be thought of as the vector \((\lambda_1, \ldots, \lambda_n)\).

Step 2. Write out the first-order conditions for the \( x_j \)s.

The first-order condition for the variable \( x_j \) \((j = 1, \ldots, k)\) is that \( \partial L/\partial x_j = 0 \), or, expanded,

\[
\frac{\partial f}{\partial x_j} - \sum_{i=1}^{n} \lambda_i \frac{\partial g_i}{\partial x_j} = 0.
\]

Step 3. Write the \( n \) constraints.

Not much to do here: The \( i \)th constraint is

\[ g_i(x) \leq c_i. \]

Step 4. Write the inequality constraints for the multipliers.

This is purely according to recipe; the multipliers must all be nonnegative, or

\[ \lambda_i \geq 0. \]

Step 5. Write the complementary slackness conditions.
A1.1. A recipe for solving problems

There are \( n \) of these — one for each constraint. The \( i \)th complementary slackness condition is that

\[
\lambda_i(c_i - g_i(x)) = 0.
\]

Stare at this one for a moment. In step 3 we required that \( g_i(x) \leq c_i \), or \( c_i - g_i(x) \geq 0 \). In step 4 we required that \( \lambda_i \geq 0 \). Hence the product \( \lambda_i(c_i - g_i(x)) \) is nonnegative. Now, in this step, we require in addition that the product be zero. That is, we require that if the \( i \)th constraint does not bind (that is, if \( c_i > g_i(x) \)), the \( i \)th multiplier must be zero. And if the \( i \)th multiplier is positive, the \( i \)th constraint must be an equality (that is, the \( i \)th constraint must bind). Note that we don’t preclude the possibility that the multiplier is zero and also that the \( i \)th constraint binds; both terms in the product could be zero.

Step 6. Mix all the above ingredients.

Look for a solution in \( x \) and \( \lambda \) to the first-order conditions, the two types of inequality constraints, and the complementary slackness conditions. When and if you find one, it is the solution to your problem.

That's it. That is the recipe for finding the solution to constrained optimization problems. For reasons alluded to below (that are properly explained in good books on optimization and nonlinear programming), this recipe will produce a solution to most of the constrained optimization problems in this book. In following this recipe, it is crucial that you get the “signs” right; this recipe works: (a) if the problem is a maximization problem; (b) if the constraints are stated in the form of \( g_i(x) \leq c_i \); (c) you are careful to write the Lagrangian precisely as above (note the minus sign in front of the sum term); and (d) you remember that the multipliers must be nonnegative.

How do you make the recipe work in variations? If you are asked to minimize \( f(x) \), then that is the same as maximizing \( -f(x) \). If you are asked to satisfy a constraint \( g_i(x) \geq c_i \), rewrite it as \( -g_i(x) \leq -c_i \). If you are asked to satisfy an equality constraint \( g_i(x) = c_i \), write it as two constraints \( g_i(x) \leq c_i \), and \( g_i(x) \geq c_i \), and then you’ll have two multipliers for these two constraints. Once you get used to adapting this basic recipe to these sorts of variations, you’ll see how to shortcut these steps. For example, you can handle equality constraints with a single multiplier if you don’t constrain that multiplier to be nonnegative and if you replace
the corresponding complementary slackness condition with the condition that the constraint must hold as an equality. But until you are used to such variations the basic recipe can be adapted, as long as you’re careful about it.

A1.2. The recipe at work: An example

That is a pretty amazing recipe. It will work, and in just a bit I’ll try to explain what it is doing and why it will work, but first let’s do an example and see it in action.

A consumer consumes two commodities, wheat and candy. If \( w \) is the amount of wheat this individual consumes and \( c \) is the amount of candy (in some units that are well specified and that we won’t bother with), then the consumer’s utility is given by the utility function \( u(w, c) = 3 \log(w) + 2 \log(c+2) \). The consumer seeks to maximize the utility he gets from his consumption of wheat and candy, subject to four constraints. The amounts of wheat and candy consumed must both be nonnegative. The consumer has $10 to spend, and the price of wheat and the price of candy are each $1 per unit. A unit of wheat contains 150 calories and a unit of candy 200 calories, and the consumer is constrained to eat no more than 1550 calories.

Mathematically, the consumer’s problem is to pick \( w \) and \( c \) to solve the following problem:

\[
\text{max } 3 \log(w) + 2 \log(2 + c) \\
\text{subject to } c \geq 0, \\
w \geq 0, \\
w + c \leq 10, \text{ and} \\
150w + 200c \leq 1550.
\]

(This mathematical formulation should be obvious to you. But in case it is not: The third constraint is the budget constraint, and the fourth is the calorie constraint.)

Now to apply the recipe from the cookbook, we rewrite the first two constraints in the form needed for our recipe: \(-w \leq 0\) and \(-c \leq 0\). Then we

1. Form the Lagrangian.
AI.2. The recipe at work: An example

I’ll use $\mu_w$ for the multiplier on the constraint $-w \leq 0$, $\mu_c$ for $-c \leq 0$, $\lambda$ for $w + c \leq 10$, and $\nu$ for $150w + 200c \leq 1550$, so the Lagrangian is

$$3 \log(w) + 2 \log(2 + c) - \lambda(w + c) - \nu(150w + 200c) + \mu_w w + \mu_c c.$$

(Note that the last term is $-\mu_c (-c)$, which comes to $\mu_c c$.)

2. Write the first-order conditions for $w$ and $c$.

They are

$$\frac{3}{w} - \lambda - 150\nu + \mu_w = 0,$$
$$\frac{2}{2 + c} - \lambda - 200\nu + \mu_c = 0.$$

3. Write the four constraints.

We’ve already done this, so I won’t repeat it here.

4. Constrain all the multipliers to be nonnegative.

$$\lambda \geq 0, \nu \geq 0, \mu_w \geq 0, \text{ and } \mu_c \geq 0.$$

5. Write the four complementary slackness conditions.

These are

$$\mu_w w = 0, \mu_c c = 0, \lambda(10 - w - c) = 0, \text{ and } \nu(1550 - 150w - 200c) = 0.$$

6. Look for a solution (in $w$, $c$, and the four multipliers) to all the equations and inequalities above.

The way this is done, typically, is by trial and error, working with the complementary slackness conditions (hereafter, CSCs). What the CSCs tell you is that either one thing or another equals zero. For example, $\mu_w w = 0$ means that either $\mu_w$ or $w$ is zero. Could it be $w$? If $w = 0$, then the first of the first-order conditions won’t be solved, since that has a $3/w$ in it. So $\mu_w$ will have to be zero, and we can disregard it in all that follows.

Next, what about $\mu_c c = 0$? Is it possible that $c = 0$? If this happens, then the second first-order condition reads $1 - \lambda - 200\nu + \mu_c = 0$. So either
\[ \lambda > 0 \text{ or } \nu > 0 \text{ or both. (Otherwise, the left-hand side of the equation just given will be at least 1; remember that multipliers are always nonnegative in this recipe.) Now by complementary slackness, the only way that } \lambda \text{ can be positive is if } w + c = 10, \text{ and since we're hypothesizing that } c = 0, \text{ this will mean } w = 10. \text{ That's not bad, since then the diet constraint is met easily: } 150 \cdot 10 + 200 \cdot 0 < 1550. \text{ But then } \nu \text{ will have to be zero. (Why? complementary slackness again.) And then the FOCs (first-order conditions), with } w = 10 \text{ and } c = 0 \text{ inserted, will read}
\]

\[ .3 - \lambda = 0 \text{ and } 1 - \lambda + \mu_c = 0. \]

The first tells us that \( \lambda = .3 \). But then the second won't work, because \( \mu_c \) will have to be \(-.7\), and negative multipliers aren't allowed. So (summing up what we know so far), if \( c = 0 \), then \( \lambda \) can't be positive. But then \( \nu \) will have to be positive (if \( c = 0 \)), and that is even worse; complementary slackness says that the only way \( \nu > 0 \) is if \( 150w + 200c = 1550 \), and this together with \( c = 0 \) gives \( w = 31/3 > 10 \), and the budget constraint is violated. So, we conclude \( c = 0 \) won't be a solution and \( \mu_c = 0 \) will have to hold.

Now look again at the first FOC: \( 3/w - \lambda - 150\nu = 0 \). I've taken the liberty of getting rid of the \( \mu_w \), since we know there won't be a solution with \( \mu_w \neq 0 \). Since \( 3/w > 0 \) for any positive \( w \), one or both of \( \lambda \) and \( \nu \) must be positive. (A similar conclusion would have emerged if we had looked at the second FOC.) So we have three cases left to try: \( \lambda > 0 \) (alone), \( \nu > 0 \) (alone), or both. Take \( \lambda > 0 \) and \( \nu = 0 \) first. Then (complementary slackness, one more time) \( w + c = 10 \). The FOCs become \( 3/w - \lambda = 0 \) and \( 2/(2 + c) - \lambda = 0 \). That is three equations in three unknowns, and we can solve \( \lambda = 3/w = 3/(10 - c) \), which substituted into the second FOC is \( 2/(2 + c) = 3/(10 - c) \), or \( (2 + c)/2 = (10 - c)/3 \), or \( 3(2 + c) = 2(10 - c) \), or \( 6 + 3c = 20 - 2c \), or \( 5c = 14 \), or \( c = 14/5 \), which then gives \( w = 10 - 14/5 = 36/5 \), and (if anyone cares) \( \lambda = 15/36 = 5/12 \).

Bingo! We have an answer! Or do we? The FOCs are solved. The multipliers are nonnegative. The CSCs hold. But we have to go back and check the original constraints. And, when we look at the calorie constraint ... disaster: \( (36/5)(150) + (14/5)(200) = 1640 \). Oh well.

Let me try \( \lambda = 0 \) and \( \nu > 0 \) next. Then by the CSC, \( 150w + 200c = 1550 \). In addition, the FOCs now read \( 3/w - 150\nu = 0 \) and \( 2/(2 + c) - 200\nu = 0 \). That is three equations in three unknowns, and we can solve. First write the two FOCs as \( 3/w = 150\nu \) and \( 2/(2 + c) = 200\nu \). Then we can eliminate \( \nu \), since \( 3/(150w) = \nu = 2/(200(2 + c)) \). Rewrite this equation as \( 150w/3 = 200(2 + c)/2 \), or \( 50w = 200 + 100c \). Multiply both sides by \( 3 \),
A1.2. The recipe at work: An example

to get $150w = 600 + 300c$, and then use this and the first equation to eliminate $w$, or $600 + 300c + 200c = 1550$, or $c = 950/500 = 1.9$. This gives $w = 4 + 3.8 = 7.8$. Which is nice, because this means this solution meets the budget constraint. It hits the calorie constraint on the nose (since it was designed to do so). It solves the FOCs (if we figure out what $\nu$ must be) and the CSCs. But we still need to check that the value of $\nu$ that comes out of the FOCs is nonnegative. And it is: $\nu = 1/(50w) = 1/390 = .00256410\ldots$

Let me summarize this solution:

$$w = 7.8, c = 1.9, \nu = .00256410\ldots,$$

with $\lambda = \mu_w = \mu_c = 0$.

There is actually one more thing to try, namely $\lambda > 0$ and $\nu > 0$. I'll leave it to you to show that this doesn't work with a few hints. First, if both multipliers are to be positive, then complementary slackness says that both constraints must be satisfied with equality, or $w + c = 10$ and $150w + 200c = 1550$. That is two equations in two unknowns, and you can solve for $w$ and $c$ directly. Then with these two values for $w$ and $c$, the two FOCs become two equations in two unknowns, those unknowns being $\lambda$ and $\nu$. And you can solve those. And when you do, one of the two multipliers will turn out to be negative — no solution there. (For those readers who have seen this all before and are reading along only to review this, you know that the solution lies where the calorie constraint and not the budget constraint binds. So you should be able to tell before doing the algebra which of the two multipliers will turn out to be negative at the point where both constraints bind.)

We've solved the problem, and in a moment I'll try to give you some intuition about just why this recipe works. But before doing so, I want to make two comments about the problem. First, if we had used a little common sense, we could have saved a bit of work. This is a very typical consumer budgeting problem, with the not-so-typical addition of a calorie constraint. Nonetheless, we can draw the shape of the feasible set of wheat-candy pairs as in figure A1.

The line b-b is the budget constraint line, and c-c is the calorie constraint line, so the shaded quadrilateral is the set of feasible consumption pairs. Now this consumer is locally insatiable; his utility increases if we increase either his wheat consumption or his candy consumption or both, so the only possible places an answer could be found would be at one of the points marked $x$, $y$, and $z$, or on the line segment $xy$, or the segment $yz$. Each of these corresponds to a particular set of binding constraints, and
so a particular set of (possibly) nonnegative multipliers, according to complementary slackness. In this particular case, the picture won’t cut down a lot on the algebra needed to find the solution, but in other applications, knowing how to draw this sort of figure and relate it to sets of binding constraints (and nonnegative multipliers) can be a substantial help. (Imagine, for example, that I added a third constraint on the amount of cholesterol that the consumer can take in. Then knowing which pairs of constraints can simultaneously bind can be a big help.)

Second, one thing we didn’t do is compute the consumer’s utility at the optimal solution given above. It is

\[ 3 \log(7.8) + 2 \log(2 + 1.9) = 8.88432431 \ldots \]

Now let me suppose that this particular consumer, having been something of a success at his diet, is allowed by his doctor to increase his caloric intake by one calorie to 1,551 per day. If we go back and solve the problem, we find that the new optimal levels of wheat and candy are \( c = 951/500 = 1.902 \) and \( w = 1951/250 = 7.804 \). This, in turn, gives him utility \( 3 \log(7.804) + 2 \log(2 + 1.902) = 8.88688776 \ldots \). So the additional utility that he gets from one extra calorie is .00256345 \ldots, or, to the third significant decimal place, the value of the multiplier \( \nu \). An amazing coincidence? No — read on.

### A1.3. Intuition

The intuition behind this recipe is most easily communicated in the special case where \( k = 1 \); that is, where the vector \( x \) is one-dimensional. Of course, this special case hides some of the subtleties. But if you see
why things work in this special case, then it shouldn’t be too hard to imagine that it will work in general — the principles are exactly the same — and you should be able to consult a book on optimization and nonlinear programming to see this done more carefully and in greater generality.

Moreover, I’m going to make a number of assumptions that in total mean that the problem we’re trying to solve is “well behaved.” To be precise, I’m going to assume that the problem we’re trying to solve is

\[
\max f(x) \text{ subject to } g_1(x) \leq c_1 \text{ and } g_2(x) \leq c_2.
\]

I’m going to assume throughout that the three functions given are differentiable everywhere, that the set of \( x \) satisfying the two constraints is a finite union of disjoint intervals, that there is no point \( x \) at which both constraints bind simultaneously, and that if \( g_i(x) = c_i \) at some point \( x \), then the derivative of \( g_i \) at this point is either strictly positive or strictly negative. (If you are reviewing this material after encountering it previously, you may recognize that I have just assumed the great-granddaddy of all constraint qualifications.)

**A review of unconstrained optimization**

But before we take on this problem in constrained optimization, we review unconstrained optimization. Suppose that the problem was to maximize \( f(x) \) without any constraints at all. Then (this is the part you’re supposed to know) you find the optimum by setting the derivative of \( f \) to zero. This is called the *first-order condition*, and it is a necessary condition for finding a maximum value of \( f \). Why? Because if at a point \( x^0 \) the derivative of \( f \) is positive, then by increasing \( x \) a bit from \( x^0 \) you’ll get a bigger value of \( f \). And if the derivative of \( f \) is negative at \( x^0 \), then \( f \) gets bigger if you decrease \( x \) a bit from \( x^0 \).

You may also recall that finding all the points where \( f' = 0 \) generates all candidates for the optimum, but there is generally no guarantee that any solution of this FOC is even locally a maximum. These solutions, called *critical points*, could be minima. Or they could be points like the point 0 in the function \( x^3 \), which are neither local minima nor maxima. To know that we have a local maximum, we check the second-order condition: If \( f \) is twice continuously differentiable and \( f'' < 0 \) at a point where \( f' = 0 \), then that point is a local maximum. (Why? If you know how to create Taylor’s series expansions, you should be able to see why the condition \( f'' < 0 \) is sufficient for a local maximum. Note that \( f'' \leq 0 \) is necessary for this, but that if \( f'' = 0 \), we need to look at higher order derivatives, assuming \( f \) has them.)
And even if we have a local maximum, there is no guarantee in general that it is a global max. This is, however, guaranteed if \( f \) is concave. Then any local maximum is a global maximum. Why? Suppose the function \( f \) is concave. That is, \( f(ax + (1-a)y) \geq af(x) + (1-a)f(y) \) for all \( x, y \) and \( 0 \leq a \leq 1 \). Then if \( x \) isn't a global max, there is some value \( y \) with \( f(y) > f(x) \). But then \( f(ax + (1-a)y) \geq af(x) + (1-a)f(y) > f(x) \) for all \( a \) greater than zero and less than one. But this says that \( x \) isn't a local max, since for \( a \) arbitrarily close to (and less than) one, \( ax + (1-a)y \) is arbitrarily close to \( x \). Moreover, if \( f \) is concave, any solution of \( f' = 0 \) is automatically a local and hence a global max. (If you never knew this before, just take my word for it.) So if \( f \) is concave, we know that satisfaction of the FOC is necessary and sufficient for a global maximum.

**Drawing pictures**

Now back to our simple example of constrained optimization: maximize \( f(x) \) subject to \( g_1(x) \leq c_1 \) and \( g_2(x) \leq c_2 \). Recall that we assumed that the set of \( x \) satisfying the two constraints is a finite union of disjoint intervals. In particular, assume that the set of \( x \) satisfying the two constraints is a union of three intervals, the middle one of which we will call \( I \). That is, the feasible set is like the shaded region in figure A2. Now what sort of conditions must hold at some point \( x^* \) in the interval \( I \), if \( x^* \) is to be a candidate solution for the maximizing value of \( x \)? (The fact that \( I \) is the middle of the three intervals is completely irrelevant to what follows.)

Consider first a case where \( x^* \) is in the interior of \( I \), as in figure A3(a). Then, just as in the case of unconstrained maximization, it had better be that \( f'(x^*) = 0 \). Why? Because at such an \( x^* \), we can move up a bit and

---

*Figure A2. The feasible set.*
Figure A3. Possible locations of the optimal solution.
In (a), the solution, \( x^* \) is in the interior of a feasible interval. In (b), it is at a right-hand endpoint, and in (c) it is at a left-hand endpoint.

...down a bit and not violate either constraint (that is, stay inside of \( I \)). This means that if \( x^* \) is going to be optimal, it has to be as good as values just above and just below it, and, as before, this implies that \( f \) cannot have a strictly positive or negative derivative at \( x^* \).

Now suppose that \( x^* \) is up against the right-hand side of \( I \), as in figure A3(b). Now \( x^* \) could be the optimal value of \( x \) even if \( f''(x^*) > 0 \);
we can increase the value of \( f \) by moving up a bit further, but this will violate one of the constraints.

And in figure A3(c) we see the other "boundary" possibility. If \( x^* \) is up against the left-hand side of the interval \( I \), then \( f'(x^*) \) could be negative and still \( x^* \) is a candidate for the optimum.

**The FOCs, the CSCs, and the preceding pictures**

Now we relate these three pictures to the FOCs and the CSCs. Let \( \lambda_i \) be the multiplier on the constraint \( g_i(x) \leq c_i \) for \( i = 1, 2 \). Then the Lagrangian is \( f(x) - \lambda_1 g_1(x) - \lambda_2 g_2(x) \), and the FOC is

\[
f'(x) - \lambda_1 g'_1(x) - \lambda_2 g'_2(x) = 0.
\]

**Case a.** Suppose we find a solution \( x^* \) to the FOC, the CSCs (and the constraints, including the nonnegativity constraints on the multipliers), such that neither constraint binds at \( x^* \). That is, \( x^* \) is in the interior of an interval such as \( I \), as in figure A3(a). Since \( x^* \) is in the interior of \( I \), the CSCs mandate that \( \lambda_1 = \lambda_2 = 0 \), and the FOC will read that \( f'(x^*) = 0 \). Precisely what we said we'd want in such a case.

**Case b.** Suppose that \( x^* \) is a solution to all the equations and constraints, the first constraint binds at \( x^* \) (and the second is slack; recall our assumptions), and \( g'_1(x^*) > 0 \). Then \( x^* \) must be on the right-hand side of an interval such as \( I \), as in figure A3(b). Why? Because since \( g'_1(x^*) > 0 \), increasing a bit past \( x^* \) will cause the constraint \( g_1(x) \leq c_1 \) to be violated, while decreasing a bit will cause this constraint to go slack (and since the other constraint is slack by assumption, it won't be a problem, for small decreases in \( x \), at least). Now the CSC allows \( \lambda_1 \) to be nonnegative; \( \lambda_2 = 0 \) must hold, however. Hence the FOC is now

\[
f'(x^*) - \lambda_1 g'_1(x^*) = 0.
\]

Since \( \lambda_1 \) is constrained to be nonnegative and \( g'_1(x^*) > 0 \) by assumption, the FOC reads that \( f'(x^*) \geq 0 \), which is precisely what we want for a candidate solution at the right-hand side of an interval; cf. figure A3(b).

**Case c.** Suppose the first constraint binds at \( x^* \), the second is slack, and \( g'_1(x^*) < 0 \). Then by logic just as above, \( x^* \) will have to be on the left-hand side of an interval like \( I \), complementary slackness will say that the FOC simplifies to \( f'(x^*) - \lambda_1 g'_1(x^*) = 0 \), and since \( \lambda_1 \geq 0 \) and \( g'_1(x^*) < 0 \), we have that \( f'(x^*) \leq 0 \), or just what we want for figure A3(c).
A few words about the coincidence in the example

About that coincidence, notice that in case b, in the solution to the FOC we'll find that $\lambda_1 = f'(x^*)/g_1'(x^*)$. Now suppose we increase the right-hand side of the constraint $g_1(x) \leq c_1$ by just a bit — say, to $c_1 + \epsilon$. This is going to lengthen the interval $I$ by a bit; the interval gets longer by approximately $\epsilon/g_1'(x^*)$. A picture may help you see this. In figure A4, we have a more detailed look at the situation in case b. On the bottom set of axes we've plotted $g_1$, and you'll see that the right-hand side of the interval $I$ is determined by the solution of $g_1(x) = c_1$. When we increase $c_1$ by $\epsilon$, we can push over the right-hand side, and to a first-order approximation, the right-hand side moves up by $\epsilon/g_1'(x^*)$. But then on the top set of axes, we see that loosening the constraint by this amount allows the function $f$ to rise a bit more; it rises by approximately the amount $f'(x^*) \times (\epsilon/g_1'(x^*))$, which is $\lambda_1 \epsilon$. That is, the rate of increase in the objective function value at our "candidate solution" per unit increase in the right-hand side of

\[ f(x^*) + \epsilon f'(x^*)/g_1'(x^*), \text{ approximately} \]

\[ f(x^*) \]

\[ f \]

\[ c_1 + \epsilon \]

\[ c_1 \]

\[ g_1 \]

\[ x^* + \epsilon/g_1'(x^*), \text{ approximately} \]

\[ x^* \]

Figure A4. Explaining the coincidence.
the binding constraint is precisely the value of the multiplier on that constraint. This isn't special to case b; you could draw a similar picture for case c. (It might be a useful exercise to do so.) And this isn't anything special to the case of one dimension either, as long as the problem you're looking at is well behaved.

Some general comments

All this talk proves nothing. It is meant to indicate how the CSCs work — by modifying the FOCs in the "natural" way when you are at a point that is up against a constraint. The pictures are harder to draw when you move from one-dimensional to multidimensional problems, and I won't try to draw them here, but the same basic principles apply; complementary slackness says that a multiplier can be strictly positive only when a constraint binds, and then that multiplier is used to allow the objective function \( f \) to increase in directions that owing to the binding constraint(s) are infeasible.

Finally, you'll note that everything discussed in this section on intuition (after the review of unconstrained optimization) was about producing candidate solutions. Just as in unconstrained optimization, you might wonder what guarantee you have that a solution to all the equations and inequalities is really globally (or even locally) optimal. Accompanying theory about things analogous to the second-order conditions in unconstrained optimization gives you sufficient conditions for a candidate solution to be locally optimal. From this I happily spare you. And a theory analogous to the result in unconstrained optimization about concave \( f \) ensures that any candidate solution is in fact globally optimal. That theory is actually fairly simple, but I'll spare you that as well. However, I will assure you that for purposes of the problems in this book, at least until we get to oligopoly, any candidate solution is in fact a global optimum. And I'll even tell you how to watch for the exceptions in general: Watch for problems where the objective function \( f \) is not concave and/or the set of solutions of the constraints is not convex.

A1.4. Bibliographic notes

The cookbook recipe and very brief intuitive remarks presented here are certainly no substitution for learning properly about constrained optimization. Because constrained optimization plays such an important role in economic theory, the serious student of economics should study (either independently or in a course on mathematical methods) this subject. At a minimum, you should know about the FOCs and CSCs as necessary
A1.5. Problems

conditions for an optimum and so-called constraint qualifications, second-order conditions, sufficiency of the FOCs and CSCs in convex problems, and problems with quasi-convex or -concave objective functions. Any number of books provide treatments of this material; Mangasarian (1969) covers much of the nuts and bolts, although this book doesn’t explain why the multipliers are partial derivatives of the value function. Luenberger (1984) does discuss this and provides much of what one needs for convex problems but doesn’t discuss quasi-convexity. Luenberger also provides an excellent treatment of linear programming and numerical methods for solving constrained and unconstrained programming problems.

References


A1.5. Problems

1. What is the solution to the dieting consumer’s problem if his calorie constraint is 1,650 calories? If his calorie constraint is 550 calories? If his calorie constraint is 1,600 calories?

2. Suppose that our consumer also has a problem with his cholesterol levels. Each unit of wheat supplies 10 unit of cholesterol, whereas each unit of candy supplies 100. The consumer has been told by his doctor to limit his cholesterol intake to no more than 260 units per day. What is the optimal solution to the consumer’s problem now, with this additional constraint (with a calorie constraint of 1,550 calories per day)? Be sure to compute the value of the multipliers. Then work through the solution if the consumer (a) is allowed 261 units of cholesterol per day, 1,550 calories per day, and has $10 to spend; (b) is allowed 260 units of cholesterol per day, 1,551 calories per day, and has $10 to spend; and (c) is allowed 260 units of cholesterol per day, 1,550 calories per day, and has $10.10 to spend. In terms of the coincidence we observed before, what happens in this case?

3. I’ve assured you that for most of the problems you encounter in this book, when you have one candidate solution, you will know that you have a global optimum. A part of this assurance may be provided by the following fact: If, in the general problem with which this appendix began, the function \( f \) is strictly concave, and the functions \( g_i \) are all convex, then the problem necessarily has at most a single solution. Prove this.