Chapter 4

Complementarity and Optimization

In a complementarity problem, one is given a function f from $\Re^n$ to $\Re^n$ and two n-vectors a and b, and asked to compute an n-vector x that satisfies

$$a_i \leq x_i \leq b_i$$

$$x_i > a_i \Longrightarrow f_i(x) \geq 0$$

$$x_i < b_i \Longrightarrow f_i(x) \leq 0$$

for all $i = 1, 2, \ldots, n$. We write the complementarity problem in abbreviated form:

$$f(x) \perp a \leq x \leq b.$$

In some applications, x is bounded only on one side. If x is unbounded below, we write $f(x) \perp x \leq b$; if x is unbounded above, we write $f(x) \perp x \geq a$. If x is unbounded in both directions, the complementarity problem reduces to a standard rootfinding problem $f(x) = 0$.

Complementarity problems arise naturally when economic variables are subject to bounds. Consider, for example, the single-good static price equilibrium model commonly encountered in introductory economics courses. In the model, equilibrium price p is characterized by equality of quantity supplied $S(p)$ and quantity demanded $D(p)$. The equilibrium price is therefore the root of the excess demand function $E(p) = D(p) - S(p)$. Suppose, however, that the government imposes a price ceiling $\bar{p}$, which it enforces through

fiat or direct market intervention. Then it is possible for excess demand to be positive in equilibrium, but only if price has hit the ceiling. In the presence of a price ceiling, computation of equilibrium price is not a rootfinding problem, but rather a complementarity problem:

$$E(p) \perp p \leq \bar{p}.$$

Most complementarity problems encountered in economics and finance have natural interpretations as conditions for an arbitrage-free equilibrium. In such applications, x is a vector of economic activities and $f(x)$ is the vector of marginal profits for each of the activities. If $f_i(x)$ is positive, then profits may be increased by raising the level of activity x_i, unless x_i is at its upper bound b_i. In $f_i(x)$ is negative, then profits may be increased by lowering the level of activity x_i, unless x_i is at its lower bound a_i. Arbitrage profit opportunities are nonexistent, that is, an arbitrage-free equilibrium exists, if, and only if, x solves the complementarity problem $f(x) \perp a \leq x \leq b$.

Another problem that arises often in economic applications is the finite-dimensional constrained optimization problem. In the constrained optimization problem one is given a real-valued function f on $\Re^n$ and asked to find its maximum (or minimum) subject to a series of constraints, as in:

$$\max_{a \leq x \leq b} f(x).$$

The constrained optimization problem is very closely related to the complementarity problem. By the Karush-Kuhn-Tucker theorem, a constrained optimum must satisfy certain complementarity conditions. These conditions typically admit an arbitrage-free equilibrium interpretation.

Complementarity and constrained optimization problems can also arise in more complicated economic models. For example, a finite-dimensional constrained optimization problem is often embedded within the Bellman functional equation that characterizes the dynamic optimum. If one solves the Euler functional equation of a dynamic optimization problem using collocation methods, one can encounter a complementarity problem if the optimal action is subject to constraints. Complementarity problems can also arise in computational procedures when the economic variables are not subject to bounds or when existing bounds are known a priori to be nonbinding at the solution. Even when variables are unbounded, it is not uncommon for an analyst to introduce artificial bounds on variables to preclude the iterates generated by the solution algorithm from straying into regions in which the

underlying objective or arbitrage profit functions are undefined or poorly behaved.

The existence of bounds make complementarity and constrained optimization problems fundamentally more difficult to solve than rootfinding and unconstrained optimization problems. Complementarity and constrained optimization problems, however, have been actively researched by numerical analysts for many years. Today, a variety of algorithms exist for solving linear complementarity problems. Nonlinear complementarity problems can be solved iteratively by reducing them to a sequence of linear complementarity problems. Constrained optimization problem may be solved by deriving their first-order necessary conditions and converting them into complementarity problems.

4.1 Linear Complementarity

In a linear complementarity problem, one is given an n by n matrix M, and n-vectors q, a, and b, and asked to compute an n-vector x that satisfies

$$M \star x + q \perp a \leq x \leq b.$$

Consider first the univariate linear complementarity problem and, for the sake of discussion, think of $mx + q$ as measuring the unit profit from some activity whose level is x. The case $m < 0$ is illustrated in figure 4.1. We consider three subcases. If $mx + q > 0$ everywhere on $[a, b]$, then there is always an incentive to increase x and the unique solution is to raise x to its maximum allowable value b. If $mx + q < 0$ everywhere on $[a, b]$, on the other hand, then there is always an incentive to decrease x and the unique solution is to lower x to its minimum allowable value a. If $mx + q$ can be either positive and negative throughout $[a, b]$, then there will be a single point in the interior of $[a, b]$ at which $mx + q = 0$. This point is the unique solution to the linear complementarity problem. It is a "stable" solution in the sense that for levels of x near the solution, the profit incentives are to move toward the solution.

Figure 4.1: Univariate Linear Complementarity, $m < 0$

The case $m > 0$ is illustrated in figure 4.2. As before, if $mx + q > 0$ everywhere on $[a, b]$ the unique solution is to increase x to its maximum allowable value b. And if $mx+q < 0$ everywhere on $[a, b]$, the unique solution is to decrease x to its minimum allowable value a. If $mx + q$ can be both positive and negative on $[a, b]$, there will three solutions. The lower bound $x = a$ is one solution, since for x slightly above a, there is an incentive to decrease x. The upper bound $x = b$ is a second solution, since for x slightly below b there is an incentive to increase x. There is a point in the interior of $[a, b]$ at which $mx + q = 0$. This point is a third solution to the linear complementarity problem. It is an "unstable" solution in the sense that for levels of x just off the solution, the incentive is to move away from the solution until a bound is encountered.

Figure 4.2: Univariate Linear Complementarity, $m > 0$

Thus, if $m < 0$, the univariate linear complementarity problem is "well-behaved", in the sense that it always has an unique, stable solution. If $m > 0$, on the other hand, the problem may possess multiple and unstable solutions. Fortunately, in most economic and dynamic equilibrium models, the profitability of an activity usually decreases with the level of the activity. In other words, in most economic equilibrium models, $m < 0$ and the equilibrium is well-defined.

Establishing the existence and uniqueness of solutions for multivariate linear complementarity problems is a bit more complicated. Generally, existence and uniqueness can be guaranteed only if the M matrix satisfies some condition that is a multivariate generalization of negativity. For example, a solution is known to exist for the linear complementarity problem if $a \leq b$ and either

- M is negative semidefinite, that is, $x' \star M \star x \leq 0$ for all x;
- M is strictly co-negative, that is, $x' \star M \star x \leq 0$ for all $x \geq 0$, $x \neq 0$.

An unique solution exists for the linear complementarity problem if $a \leq b$ and either

- M is negative definite, that is, $x' \star M \star x < 0$ for all $x \neq 0$;

- M is an N-matrix, that is, the principal minors of $-M$ are all positive;
- M is diagonally dominant, that is, $|M_{ii}| > \sum_{j \neq i} |M_{ij}|$ and $M_{ii} < 0$ for all i.

Perhaps the simplest way to solve a linear complementarity problem is to test all "basic" vectors. A vector x if basic if it satisfies the necessary, but not sufficient, conditions for a solution of the linear complementarity problem that, for every $i = 1, 2, \ldots, n$, either $x_i = a_i$, $x_i = b_i$, or $z_i = 0$, where $z = M \star x + q$. If all principal minors of M are invertible, then there can be at most 3^n basic vectors. Thus, if n is small, one could easily solve the linear complementarity problem by enumerating all the basic vectors and testing to see which, if any, satisfy the full complementarity conditions.

More specifically, to solve a linear complementarity problem by complete enumeration, one sifts systematically through all 3^n tripartite partitions (α, β, γ) of the index set $\{1, 2, \ldots, n\}$, computing, for each partition, the associated basic vector x, which is given by

$$\begin{aligned} x_\alpha &= a_\alpha, \\ x_\beta &= b_\beta, \\ x_\gamma &= -M_{\gamma\gamma}[M_{\gamma\alpha} a_\alpha + M_{\gamma\beta} b_\beta + q_\gamma], \end{aligned}$$

If the basic vector satisfies $z_\alpha \leq 0$, $z_\beta \geq 0$, and $a_\gamma \leq x_\gamma \leq b_\gamma$, then it is a solution to the linear complementarity problem.

Complete enumeration has some desirable properties for a linear complementarity solution algorithm. First, if the complementarity problem has a solution, the algorithm is guaranteed to find it. Second, if the complementarity problem has more than one solution, the algorithm will find all such solutions if allowed to test all basic vectors. Unfortunately, the complete enumeration algorithm is not practical if n is large. However, for moderate to large n, there exists other algorithms based on pivoting that will be substantially faster.

The Baard principal pivoting algorithm is an relatively simple example of a linear complementarity algorithm. The Baard algorithm is an iterative procedure. Given the current iterate x, the subsequent iterate is the basic vector associated with the partition (α, β, γ), where $\alpha = \{i | w_i \leq a_i\}$, $\beta = \{i | w_i \geq b_i\}$, and $\gamma = \{i | a_i < w_i < b_i\}$, where $w = x + M \star x + q$. If an iterate remains unchanged in two consecutive iterations, it must solve the linear complementarity problem. The following code captures the essence of the Baard algorithm:

```
for it=1:maxit
   w = x+M*x+q;
   i = find(w<=a);
   j = find(a<w & w<b);
   k = find(w>=b);
   x(i) = a(i);
   x(k) = b(k);
   x(j) = -M(j,j)\(q(j)+M(j,i)*x(i)+M(j,k)*x(k));
   z = M*x+q;
   if all(z(i)<=0)&all(z(k)=>0)&(a<=x)&(x<=b), return, end;
end
```

The Baard algorithm is simple and often outperforms other algorithms in the applications most commonly encountered in dynamic economic analysis. The Baard algorithm, however, is prone to cycling. Cycling occurs when the algorithm repeatedly returns to the same basic vector without converging. Cycling can be remedied, however, by keeping track of the basic vectors visited by the algorithm and, when a basic vector is encountered for a second time without converging, jumping to another, previously unvisited basic vector using an arbitrary selection rule.

The most commonly used linear complementarity algorithm is Lemke's method. Over time, Lemke's method has proven to be the fastest and safest method for solving general linear complementarity problems. For this reason, it is the algorithm most commonly employed in general purpose commercial code. Lemke's algorithm outperforms the Baard method as n becomes larger than 10 or 15. However, because the linear complementarity problems encountered in this book tend to be small, we have elected to stick with the Baard method throughout and say little more about Lemke's method.

4.2 Nonlinear Complementarity

The nonlinear complementarity problem takes the general form

$$f(x) \perp a \leq x \leq b.$$

In practice, most nonlinear complementarity problems are solved using the *Josephy-Newton method*. The Josephy-Newton method is a generalization of the Newton rootfinding method. Like the Newton method, the Josephy-Newton method employs the principle of *successive linearization*. Successive

linearization calls for the nonlinear complementarity problem to be replaced with a sequence of simpler linear complementarity problems whose solutions, under certain conditions, converge to the solution of the nonlinear problem.

The Josephy-Newton method begins with the analyst supplying a guess x_0 for the solution to the problem. Given the k^{th} iterate x_k, the subsequent iterate x_{k+1} is computed by solving the linear complementarity problem

$$L_k f(x) \perp a \leq x \leq b$$

where

$$L_k f(x) = f(x_k) + f'(x_k)(x - x_k)$$

is the Taylor linear approximation to f about x_k. Iterates are generated sequentially until the change in successive iterates becomes acceptably small.

Univariate nonlinear complementarity problems are relatively easy to solve. Assume that the user has provided an initial guess `x` for the solution, a convergence tolerance `tol`, an upper limit `maxit` on the number of iterations, and a routine `func` that computes the value `f` and derivative `d` of the function f at an arbitrary point. Then the following code segment executes the Josephy-Newton method for the univariate problem:

```
for it=1:maxit
   xold = x;
   [f,d] = func(xold);
   x = xold - d\f;
   x = max(x,a);
   x = min(x,b);
   if norm(x-xold)<tol, break, end;
end
```

The univariate Josephy-Newton method is graphically illustrated in figure 4.3. The algorithm begins with the analyst supplying a guess x_0 for the solution. The function f is approximated by its first-order Taylor series expansion about x_0, which is graphically represented by the line tangent to f at x_0. The solution x_1 to the resulting linear complementarity problem, which in this case is internal to the interval $[a, b]$, is then accepted as an improved estimate for the solution to the original nonlinear problem. In the second step, the line tangent to f at x_1 is constructed and the resulting linear complementarity problem is solved for the subsequent iterate, which

in this case will be $x_2 = b$. In the third step, the line tangent to f at x_2 is constructed and the resulting linear complementarity problem is solved for the subsequent iterate, which in this case will again be $x_3 = b$. Because third iterate is unchanged from the second, the Josephy-Newton algorithm stops, having found the solution $x = b$ to the original nonlinear complementarity problem.

Figure 4.3: Josephy-Newton method.

More generally, solving a multivariate nonlinear complementarity problems using the Josephy-Newton method requires a specialized routine for solving linear complementarity problems. Suppose the function `lcpsolve` solves the linear complementarity problem $M \star x + q \perp a \leq x \leq b$ using the call

```
x = lcpsolve(xinit,a,b,M,q);
```

where `xinit` is an initial guess for the solution. Then the nonlinear complementarity problem $f(x) \perp a \leq x \leq b$ can be solved via the Josephy-Newton method using the following code segment:

```
for it=1:maxit
   xold = x;
   [f,d] = func(xold);
   x = lcpsolve(xold,a,b,d,f-d*xold);
   if norm(x-xold)<tol, break, end;
end
```

Here, the user must provide an initial guess `x` for the solution, a convergence tolerance `tol`, an upper limit `maxit` on the number of iterations, and a routine `func` that computes the value `f` and Jacobian `d` of the function f at an arbitrary point.

Like the Newton method, the Josephy-Newton method converges if f is continuously differentiable and if the initial value of x supplied by the analyst is "sufficiently" close to a solution of the nonlinear complementarity problem. There is, however, no generally practical formula for determining what sufficiently close is. Typically, an analyst makes a reasonable guess for

the solution and counts his blessings if the iterates converge. If the iterates do not converge, then the analyst must look more closely at the analytic properties of f to find a better starting value. The Josephy-Newton method can be robust to starting value if f is well behaved, for example, if f is strictly concave. The Josephy-Newton method, however, can also be very sensitive to starting value, for example, if f has high derivatives that change sign frequently.

In practice, the most common cause of convergence failure in Josephy-Newton method is not a poor starting value, but rather a programming error by the analyst. While the Josephy-Newton method tends to be far more robust to initialization than the underlying theory suggests, particularly when f is concave, the iterates can easily explode or begin to jump around wildly if either the user-supplied function and derivative evaluation routines contain a coding error. For this reason, the analyst should always verify his or her code by comparing the derivatives computed by the derivative evaluation routine with those computed using finite differencing and the function routine. Typically, a programming error in either the function or derivative code will show up clearly in such a comparison.

An alternative to the Josephy-Newton method that does not require the explicit computation of the Jacobian is a Josephy-quasi-Newton method. A generalization of Broyden's rootfinding method to the nonlinear complementarity problem, for example, takes the form:

```
f = func(x);
for it=1:maxit
   x = xold;
   x = lcpsolve(xold,a,b,d,f-A*xold);
   delx = x - xold;
   if norm(delx)<tol, break, end;
   f = func(x);
   A = A + f*delx'/(delx'*delx);
end
```

This method, which can be called the Josephy-Broyden method, requires the analyst to provide both an initial estimate of the solution `x` and an initial guess for the Jacobian at the solution `A`. Often, `A` is initialized as the identity matrix, although using a finite-difference approximation to the Jacobian at the initial x may be a safer choice. As is the case with its rootfinding counterparts, the Josephy-Broyden method typically take more

iterations than the Josephy-Newton method, but requires less computation per iteration.

4.3 Finite-Dimensional Optimization

In the general finite-dimensional optimization problem, one is given a real-valued function f defined on $X \subset \Re^n$, and asked to find an $x^* \in X$ such that $f(x^*) \geq f(x)$ for all $x \in X$. We denote this problem

$$\max_{x \in X} f(x)$$

and call f the objective function, X the feasible set, and x^*, if it exists, an optimum. By the Theorem of Wieirstrass, if f is continuous and X is nonempty, closed, and bounded, then f has an optimum on X.

A point $x^* \in X$ is a *local maximum* of f if there is an ϵ-neighborhood N of x^* such that $f(x^*) \geq f(x)$ for all $x \in N \cap X$. The point x^* is a strict local maximum if, additionally, $f(x^*) > f(x)$ for all $x \neq x^*$ in $N \cap X$. If x^* is a local maximum of f that resides in the interior of X and f is twice differentiable there, then $f'(x^*) = 0$ and $f''(x^*)$ is negative semidefinite. Conversely, if $f'(x^*) = 0$ and $f''(x)$ is negative semidefinite in an ϵ-neighborhood of x^* contained in X, then x^* is a local maximum; if $f''(x^*)$ is negative definite, then x^* is a strict local maximum. By the Local-Global Theorem, if f is concave, X is convex, and x^* is a local maximum of f, the x^* is a global maximum of f on X.

For most optimization problems encountered in computational economics applications, the constraint set is typically characterized through a series of inequalities. The simplest constrained optimization problem is the bound-constrained optimization problem

$$\max_{a \leq x \leq b} f(x)$$

where a and b are vectors in $\Re^n$ such that $a \leq b$. According to the Karush-Kuhn-Tucker theorem, if f is differentiable on $[a, b]$, then x^* is an constrained optimum of f only if it solves the nonlinear complementarity problem

$$f'(x) \perp a \leq x \leq b.$$

Conversely, if f is concave on $[a, b]$ and x^* solves the nonlinear complementarity problem, then it is an constrained optimum of f; if additionally f is strictly concave on $[a, b]$, then the optimum is unique.

The sensitivity of the optimal value of the objective function f^* to changes in the bounds of a bound-constrained optimization problem are relatively easy to characterize. According to the Envelope theorem,

$$\begin{aligned}\frac{df^*}{da} &= \min\{0, f'(x^*)\} \\ \frac{df^*}{db} &= \max\{0, f'(x^*)\}.\end{aligned}$$

More generally, if f, a, and b all depend on some parameter p, then

$$\frac{df^*}{dp} = \frac{\partial f}{\partial p} + \min\{0, \frac{\partial f}{\partial x}\}\frac{da}{dp} + \max\{0, \frac{\partial f}{\partial x}\}\frac{db}{dp},$$

where the derivatives of f, a, and b are evaluated at (x^*, p).

One way to solve a bound-constrained optimization problem is to solve its Karush-Kuhn-Tucker complementarity conditions using the Josephy-Newton method. The analyst begins by supplying an initial guess for the optimum x_0. Given the k^{th} iterate x_k, one then computes the subsequent iterate x_{k+1} by solving the linear complementarity problem

$$f'(x_k) + f''(x_k)(x - x_k) \perp a \leq x \leq b.$$

Iterates are generated until a convergence criterion is satisfied. This approach to solving the bound-constrained optimization problem is also known as the method of *successive quadratic programming*, because it is equivalent to solving the sequence of quadratic programs

$$\max_{a \leq x \leq b} Q_k f(x)$$

where

$$Q_k f(x) = f(x_k) + f'(x_k)(x - x_k) + 0.5(x - x_k)' f''(x_k)(x - x_k)$$

is the Taylor quadratic approximation to f about x_k.

The general constrained optimization problem allows for nonlinear inequality constraints, as in

$$\begin{aligned}\max_{x \in \Re^n} \quad & f(x) \\ s.t. \quad & g(x) \leq b \\ & x \geq 0,\end{aligned}$$

where g is an arbitrary map from $\Re^n$ to $\Re^m$.

According to the Karush-Kuhn-Tucker Theorem, a regular point x^* maximizes f only if there is a vector $\lambda^* \in \Re^m$ such that x^* and λ^* satisfy the so-called Karush-Kuhn-Tucker complementarity conditions

$$\begin{aligned} f'(x^*) - \lambda^* g'(x^*) &\perp x^* \geq 0 \\ g(x^*) - b &\perp \lambda^* \geq 0. \end{aligned}$$

A point x is regular if the gradients of all constraint functions g_j that satisfy $g_j(x^*) = b_j$ are linearly independent. The condition of regularity may be omitted from the statement of the theorem if either the constraint functions are all linear, or if f is concave, the g_j are convex, and the feasible region has a nonempty interior. Conversely, if f is concave, the g_j are convex, and (x^*, λ^*) satisfy the Karush-Kuhn-Tucker conditions, then x^* solves the general constrained optimization problem.

In the Karush-Kuhn-Tucker complementarity conditions, the λ_j^* are called Lagrangian multipliers or shadow prices. The significance of the shadow prices is given by the Envelope Theorem, which asserts that under mild regularity conditions,

$$\frac{\partial f^*}{\partial b} = \lambda^*,$$

that is, λ_j^* is the rate at which the optimal value of the objective will change with changes in the right-hand-side constant b_j.

The Karush-Kuhn-Tucker complementarity conditions have a natural arbitrage interpretation. Suppose $x_1, x_2, \ldots, x_n$ are levels of certain economic activities and the objective is to maximize profit $f(x)$ generated by those activities subject to resource availability constraints of the form $g_j(x) \leq b_j$. Then λ_j^* represents the opportunity cost or shadow price of the jth resource and

$$MP_i = \frac{\partial f}{\partial x_i} - \sum_j \lambda_j^* \frac{\partial g_j}{\partial x_i}$$

represents the economic marginal profit of the ith activity, accounting for the opportunity cost of the resources used in activity i. The Karush-Kuhn-Tucker conditions may thus be interpreted as follows:

$x_i \geq 0$	Activity levels are nonnegative.
$MP_i \leq 0$	All profit opportunities eliminated.
$MP_i < 0 \Rightarrow x_i = 0$	Avoid unprofitable activities.
$\lambda_j^* \geq 0$	Shadow price of resource is nonnegative.
$g_j(x) \leq b_j$	Resource use cannot exceed availability.
$g_j(x) < b_j \Rightarrow \lambda_j = 0$	Surplus resource has no economic value.

General constrained optimization problems with nonlinear constraints are fundamentally more difficult to solve than simple bound-constrained optimization problems. Since we will encounter only bound-constrained optimization later in the book, we will not discuss general nonlinear constrained optimization algorithms. The interested reader is referred to the many good references currently available on the subject.