

# CHAPTER *17*

## Penalty and Augmented Lagrangian Methods

Some important methods for constrained optimization replace the original problem by a sequence of subproblems in which the constraints are represented by terms added to the objective. In this chapter we describe three approaches of this type. The *quadratic penalty* method adds a multiple of the square of the violation of each constraint to the objective. Because of its simplicity and intuitive appeal, this approach is used often in practice, although it has some important disadvantages. In *nonsmooth exact penalty* methods, a single unconstrained problem (rather than a sequence) takes the place of the original constrained problem. Using these penalty functions, we can often find a solution by performing a single

unconstrained minimization, but the nonsmoothness may create complications. A popular function of this type is the  $\ell_1$  penalty function. A different kind of exact penalty approach is the *method of multipliers* or *augmented Lagrangian method*, in which explicit Lagrange multiplier estimates are used to avoid the ill-conditioning that is inherent in the quadratic penalty function.

A somewhat related approach is used in the *log-barrier method*, in which logarithmic terms prevent feasible iterates from moving too close to the boundary of the feasible region. This approach forms part of the foundation for interior-point methods for nonlinear programming and we discuss it further in Chapter 19.

## 17.1 THE QUADRATIC PENALTY METHOD

### MOTIVATION

Let us consider replacing a constrained optimization problem by a single function consisting of

- the original objective of the constrained optimization problem, *plus*
- one additional term for each constraint, which is positive when the current point  $x$  violates that constraint and zero otherwise.

Most approaches define a *sequence* of such penalty functions, in which the penalty terms for the constraint violations are multiplied by a positive coefficient. By making this coefficient larger, we penalize constraint violations more severely, thereby forcing the minimizer of the penalty function closer to the feasible region for the constrained problem.

The simplest penalty function of this type is the *quadratic penalty function*, in which the penalty terms are the squares of the constraint violations. We describe this approach first in the context of the equality-constrained problem

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, \quad i \in \mathcal{E}, \quad (17.1)$$

which is a special case of (12.1). The quadratic penalty function  $Q(x; \mu)$  for this formulation is

$$Q(x; \mu) \stackrel{\text{def}}{=} f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x), \quad (17.2)$$

where  $\mu > 0$  is the *penalty parameter*. By driving  $\mu$  to  $\infty$ , we penalize the constraint violations with increasing severity. It makes good intuitive sense to consider a sequence of values  $\{\mu_k\}$  with  $\mu_k \uparrow \infty$  as  $k \rightarrow \infty$ , and to seek the approximate minimizer  $x_k$  of  $Q(x; \mu_k)$  for each  $k$ . Because the penalty terms in (17.2) are smooth, we can use techniques from

unconstrained optimization to search for  $x_k$ . In searching for  $x_k$ , we can use the minimizers  $x_{k-1}$ ,  $x_{k-2}$ , etc., of  $Q(\cdot; \mu)$  for smaller values of  $\mu$  to construct an initial guess. For suitable choices of the sequence  $\{\mu_k\}$  and the initial guesses, just a few steps of unconstrained minimization may be needed for each  $\mu_k$ .

---

□ **EXAMPLE 17.1**

Consider the problem (12.9) from Chapter 12, that is,

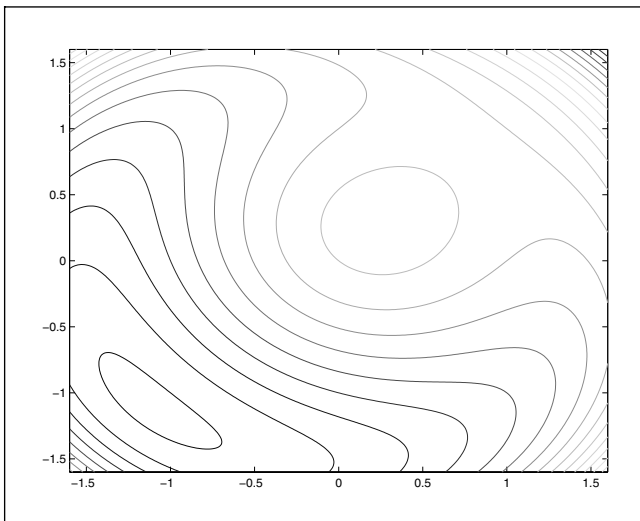
$$\min x_1 + x_2 \quad \text{subject to } x_1^2 + x_2^2 - 2 = 0, \quad (17.3)$$

for which the solution is  $(-1, -1)^T$  and the quadratic penalty function is

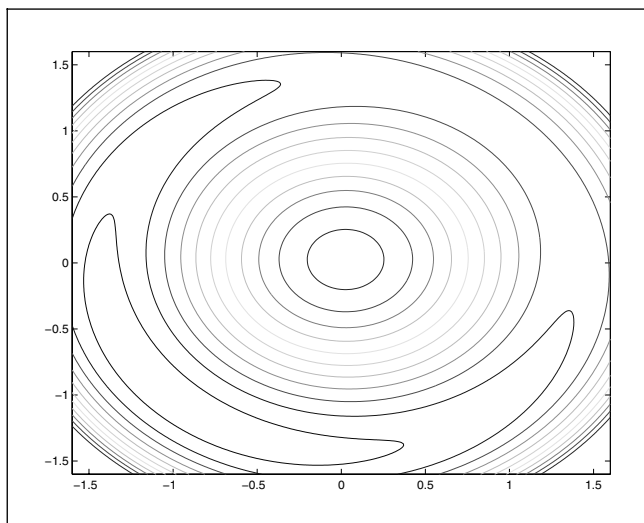
$$Q(x; \mu) = x_1 + x_2 + \frac{\mu}{2} (x_1^2 + x_2^2 - 2)^2. \quad (17.4)$$

We plot the contours of this function in Figures 17.1 and 17.2. In Figure 17.1 we have  $\mu = 1$ , and we observe a minimizer of  $Q$  near the point  $(-1.1, -1.1)^T$ . (There is also a local maximizer near  $x = (0.3, 0.3)^T$ .) In Figure 17.2 we have  $\mu = 10$ , so points that do not lie on the feasible circle defined by  $x_1^2 + x_2^2 = 2$  suffer a much greater penalty than in the first figure—the “trough” of low values of  $Q$  is clearly evident. The minimizer in this figure is much closer to the solution  $(-1, -1)^T$  of the problem (17.3). A local maximum lies near  $(0, 0)^T$ , and  $Q$  goes rapidly to  $\infty$  outside the circle  $x_1^2 + x_2^2 = 2$ . □

---



**Figure 17.1** Contours of  $Q(x; \mu)$  from (17.4) for  $\mu = 1$ , contour spacing 0.5.



**Figure 17.2** Contours of  $Q(x; \mu)$  from (17.4) for  $\mu = 10$ , contour spacing 2.

The situation is not always so benign as in Example 17.1. For a given value of the penalty parameter  $\mu$ , the penalty function may be unbounded below even if the original constrained problem has a unique solution. Consider for example

$$\min -5x_1^2 + x_2^2 \quad \text{subject to } x_1 = 1, \quad (17.5)$$

whose solution is  $(1, 0)^T$ . The penalty function is unbounded for any  $\mu < 10$ . For such values of  $\mu$ , the iterates generated by an unconstrained minimization method would usually diverge. This deficiency is, unfortunately, common to all the penalty functions discussed in this chapter.

For the general constrained optimization problem

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, \quad i \in \mathcal{E}, \quad c_i(x) \geq 0, \quad i \in \mathcal{I}, \quad (17.6)$$

which contains inequality constraints as well as equality constraints, we can define the quadratic penalty function as

$$Q(x; \mu) \stackrel{\text{def}}{=} f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \frac{\mu}{2} \sum_{i \in \mathcal{I}} ([c_i(x)]^-)^2, \quad (17.7)$$

where  $[y]^-$  denotes  $\max(-y, 0)$ . In this case,  $Q$  may be less smooth than the objective and constraint functions. For instance, if one of the inequality constraints is  $x_1 \geq 0$ , then the function  $\min(0, x_1)^2$  has a discontinuous second derivative, so that  $Q$  is no longer twice continuously differentiable.

**ALGORITHMIC FRAMEWORK**

A general framework for algorithms based on the quadratic penalty function (17.2) can be specified as follows.

**Framework 17.1** (Quadratic Penalty Method).

Given  $\mu_0 > 0$ , a nonnegative sequence  $\{\tau_k\}$  with  $\tau_k \rightarrow 0$ , and a starting point  $x_0^s$ ;

**for**  $k = 0, 1, 2, \dots$

    Find an approximate minimizer  $x_k$  of  $Q(\cdot; \mu_k)$ , starting at  $x_k^s$ ,  
    and terminating when  $\|\nabla_x Q(x; \mu_k)\| \leq \tau_k$ ;

**if** final convergence test satisfied

**stop** with approximate solution  $x_k$ ;

**end (if)**

    Choose new penalty parameter  $\mu_{k+1} > \mu_k$ ;

    Choose new starting point  $x_{k+1}^s$ ;

**end (for)**

The parameter sequence  $\{\mu_k\}$  can be chosen adaptively, based on the difficulty of minimizing the penalty function at each iteration. When minimization of  $Q(x; \mu_k)$  proves to be expensive for some  $k$ , we choose  $\mu_{k+1}$  to be only modestly larger than  $\mu_k$ ; for instance  $\mu_{k+1} = 1.5\mu_k$ . If we find the approximate minimizer of  $Q(x; \mu_k)$  cheaply, we could try a more ambitious increase, for instance  $\mu_{k+1} = 10\mu_k$ . The convergence theory for Framework 17.1 allows wide latitude in the choice of nonnegative tolerances  $\tau_k$ ; it requires only that  $\tau_k \rightarrow 0$ , to ensure that the minimization is carried out more accurately as the iterations progress.

There is no guarantee that the stop test  $\|\nabla_x Q(x; \mu_k)\| \leq \tau_k$  will be satisfied because, as discussed above, the iterates may move away from the feasible region when the penalty parameter is not large enough. A practical implementation must include safeguards that increase the penalty parameter (and possibly restore the initial point) when the constraint violation is not decreasing rapidly enough, or when the iterates appear to be diverging.

When only equality constraints are present,  $Q(x; \mu_k)$  is smooth, so the algorithms for unconstrained minimization described in the first chapters of the book can be used to identify the approximate solution  $x_k$ . However, the minimization of  $Q(x; \mu_k)$  becomes more difficult to perform as  $\mu_k$  becomes large, unless we use special techniques to calculate the search directions. For one thing, the Hessian  $\nabla_{xx}^2 Q(x; \mu_k)$  becomes arbitrarily ill conditioned near the minimizer. This property alone is enough to make many unconstrained minimization algorithms such as quasi-Newton and conjugate gradient perform poorly. Newton's method, on the other hand, is not sensitive to ill conditioning of the Hessian, but it, too, may encounter difficulties for large  $\mu_k$  for two other reasons. First, ill conditioning of  $\nabla_{xx}^2 Q(x; \mu_k)$  might be expected to cause numerical problems when we solve the linear equations to calculate the Newton step. We discuss this issue below, and show that these effects are not severe and

that a reformulation of the Newton equations is possible. Second, even when  $x$  is close to the minimizer of  $Q(\cdot; \mu_k)$ , the quadratic Taylor series approximation to  $Q(x; \mu_k)$  about  $x$  is a reasonable approximation of the true function only in a small neighborhood of  $x$ . This property can be seen in Figure 17.2, where the contours of  $Q$  near the minimizer have a “banana” shape, rather than the elliptical shape that characterizes quadratic functions. Since Newton’s method is based on the quadratic model, the steps that it generates may not make rapid progress toward the minimizer of  $Q(x; \mu_k)$ . This difficulty can be lessened by a judicious choice of the starting point  $x_{k+1}^s$ , or by setting  $x_{k+1}^s = x_k$  and choosing  $\mu_{k+1}$  to be only modestly larger than  $\mu_k$ .

### CONVERGENCE OF THE QUADRATIC PENALTY METHOD

We describe some convergence properties of the quadratic penalty method in the following two theorems. We restrict our attention to the equality-constrained problem (17.1), for which the quadratic penalty function is defined by (17.2).

For the first result we assume that the penalty function  $Q(x; \mu_k)$  has a (finite) minimizer for each value of  $\mu_k$ .

#### Theorem 17.1.

*Suppose that each  $x_k$  is the exact global minimizer of  $Q(x; \mu_k)$  defined by (17.2) in Framework 17.1 above, and that  $\mu_k \uparrow \infty$ . Then every limit point  $x^*$  of the sequence  $\{x_k\}$  is a global solution of the problem (17.1).*

PROOF. Let  $\bar{x}$  be a global solution of (17.1), that is,

$$f(\bar{x}) \leq f(x) \quad \text{for all } x \text{ with } c_i(x) = 0, \quad i \in \mathcal{E}.$$

Since  $x_k$  minimizes  $Q(\cdot; \mu_k)$  for each  $k$ , we have that  $Q(x_k; \mu_k) \leq Q(\bar{x}; \mu_k)$ , which leads to the inequality

$$f(x_k) + \frac{\mu_k}{2} \sum_{i \in \mathcal{E}} c_i^2(x_k) \leq f(\bar{x}) + \frac{\mu_k}{2} \sum_{i \in \mathcal{E}} c_i^2(\bar{x}) = f(\bar{x}). \quad (17.8)$$

By rearranging this expression, we obtain

$$\sum_{i \in \mathcal{E}} c_i^2(x_k) \leq \frac{2}{\mu_k} [f(\bar{x}) - f(x_k)]. \quad (17.9)$$

Suppose that  $x^*$  is a limit point of  $\{x_k\}$ , so that there is an infinite subsequence  $\mathcal{K}$  such that

$$\lim_{k \in \mathcal{K}} x_k = x^*.$$

By taking the limit as  $k \rightarrow \infty$ ,  $k \in \mathcal{K}$ , on both sides of (17.9), we obtain

$$\sum_{i \in \mathcal{E}} c_i^2(x^*) = \lim_{k \in \mathcal{K}} \sum_{i \in \mathcal{E}} c_i^2(x_k) \leq \lim_{k \in \mathcal{K}} \frac{2}{\mu_k} [f(\bar{x}) - f(x_k)] = 0,$$

where the last equality follows from  $\mu_k \uparrow \infty$ . Therefore, we have that  $c_i(x^*) = 0$  for all  $i \in \mathcal{E}$ , so that  $x^*$  is feasible. Moreover, by taking the limit as  $k \rightarrow \infty$  for  $k \in \mathcal{K}$  in (17.8), we have by nonnegativity of  $\mu_k$  and of each  $c_i(x_k)^2$  that

$$f(x^*) \leq f(x^*) + \lim_{k \in \mathcal{K}} \frac{\mu_k}{2} \sum_{i \in \mathcal{E}} c_i^2(x_k) \leq f(\bar{x}).$$

Since  $x^*$  is a feasible point whose objective value is no larger than that of the global solution  $\bar{x}$ , we conclude that  $x^*$ , too, is a global solution, as claimed.  $\square$

Since this result requires us to find the *global* minimizer for each subproblem, this desirable property of convergence to the global solution of (17.1) cannot be attained in general. The next result concerns convergence properties of the sequence  $\{x_k\}$  when we allow inexact (but increasingly accurate) minimizations of  $Q(\cdot; \mu_k)$ . In contrast to Theorem 17.1, it shows that the sequence may be attracted to infeasible points, or to any KKT point (that is, a point satisfying first-order necessary conditions; see (12.34)), rather than to a minimizer. It also shows that the quantities  $\mu_k c_i(x_k)$  may be used as estimates of the Lagrange multipliers  $\lambda_i^*$  in certain circumstances. This observation is important for the analysis of augmented Lagrangian methods in Section 17.3.

To establish the result we will make the (optimistic) assumption that the stop test  $\|\nabla_x Q(x; \mu_k)\| \leq \tau_k$  is satisfied for all  $k$ .

**Theorem 17.2.**

*Suppose that the tolerances and penalty parameters in Framework 17.1 satisfy  $\tau_k \rightarrow 0$  and  $\mu_k \uparrow \infty$ . Then if a limit point  $x^*$  of the sequence  $\{x_k\}$  is infeasible, it is a stationary point of the function  $\|c(x)\|^2$ . On the other hand, if a limit point  $x^*$  is feasible and the constraint gradients  $\nabla c_i(x^*)$  are linearly independent, then  $x^*$  is a KKT point for the problem (17.1). For such points, we have for any infinite subsequence  $\mathcal{K}$  such that  $\lim_{k \in \mathcal{K}} x_k = x^*$  that*

$$\lim_{k \in \mathcal{K}} -\mu_k c_i(x_k) = \lambda_i^*, \quad \text{for all } i \in \mathcal{E}, \quad (17.10)$$

where  $\lambda^*$  is the multiplier vector that satisfies the KKT conditions (12.34) for the equality-constrained problem (17.1).

PROOF. By differentiating  $Q(x; \mu_k)$  in (17.2), we obtain

$$\nabla_x Q(x_k; \mu_k) = \nabla f(x_k) + \sum_{i \in \mathcal{E}} \mu_k c_i(x_k) \nabla c_i(x_k), \quad (17.11)$$

so from the termination criterion for Framework 17.1, we have that

$$\left\| \nabla f(x_k) + \sum_{i \in \mathcal{E}} \mu_k c_i(x_k) \nabla c_i(x_k) \right\| \leq \tau_k. \quad (17.12)$$

By rearranging this expression (and in particular using the inequality  $\|a\| - \|b\| \leq \|a + b\|$ ), we obtain

$$\left\| \sum_{i \in \mathcal{E}} c_i(x_k) \nabla c_i(x_k) \right\| \leq \frac{1}{\mu_k} [\tau_k + \|\nabla f(x_k)\|]. \quad (17.13)$$

Let  $x^*$  be a limit point of the sequence of iterates. Then there is a subsequence  $\mathcal{K}$  such that  $\lim_{k \in \mathcal{K}} x_k = x^*$ . When we take limits as  $k \rightarrow \infty$  for  $k \in \mathcal{K}$ , the bracketed term on the right-hand-side approaches  $\|\nabla f(x^*)\|$ , so because  $\mu_k \uparrow \infty$ , the right-hand-side approaches zero. From the corresponding limit on the left-hand-side, we obtain

$$\sum_{i \in \mathcal{E}} c_i(x^*) \nabla c_i(x^*) = 0. \quad (17.14)$$

We can have  $c_i(x^*) \neq 0$  (if the constraint gradients  $\nabla c_i(x^*)$  are dependent), but in this case (17.14) implies that  $x^*$  is a stationary point of the function  $\|c(x)\|^2$ .

If, on the other hand, the constraint gradients  $\nabla c_i(x^*)$  are linearly independent at a limit point  $x^*$ , we have from (17.14) that  $c_i(x^*) = 0$  for all  $i \in \mathcal{E}$ , so  $x^*$  is feasible. Hence, the second KKT condition (12.34b) is satisfied. We need to check the first KKT condition (12.34a) as well, and to show that the limit (17.10) holds.

By using  $A(x)$  to denote the matrix of constraint gradients (also known as the Jacobian), that is,

$$A(x)^T = [\nabla c_i(x)]_{i \in \mathcal{E}}, \quad (17.15)$$

and  $\lambda_k$  to denote the vector  $-\mu_k c(x_k)$ , we have as in (17.12) that

$$A(x_k)^T \lambda_k = \nabla f(x_k) - \nabla_x Q(x_k; \mu_k), \quad \|\nabla_x Q(x_k; \mu_k)\| \leq \tau_k. \quad (17.16)$$

For all  $k \in \mathcal{K}$  sufficiently large, the matrix  $A(x_k)$  has full row rank, so that  $A(x_k)A(x_k)^T$  is nonsingular. By multiplying (17.16) by  $A(x_k)$  and rearranging, we have that

$$\lambda_k = [A(x_k)A(x_k)^T]^{-1} A(x_k) [\nabla f(x_k) - \nabla_x Q(x_k; \mu_k)].$$

Hence by taking the limit as  $k \in \mathcal{K}$  goes to  $\infty$ , we find that

$$\lim_{k \in \mathcal{K}} \lambda_k = \lambda^* = [A(x^*)A(x^*)^T]^{-1} A(x^*) \nabla f(x^*).$$



By taking limits in (17.12), we conclude that

$$\nabla f(x^*) - A(x^*)^T \lambda^* = 0, \quad (17.17)$$

so that  $\lambda^*$  satisfies the first KKT condition (12.34a) for (17.1). Hence,  $x^*$  is a KKT point for (17.1), with unique Lagrange multiplier vector  $\lambda^*$ .  $\square$

It is reassuring that, if a limit point  $x^*$  is not feasible, it is at least a stationary point for the function  $\|c(x)\|^2$ . Newton-type algorithms can always be attracted to infeasible points of this type. (We see the same effect in Chapter 11, in our discussion of methods for nonlinear equations that use the sum-of-squares merit function  $\|r(x)\|^2$ .) Such methods cannot be guaranteed to find a root, and can be attracted to a stationary point or minimizer of the merit function. In the case in which the nonlinear program (17.1) is infeasible, we often observe convergence of the quadratic-penalty method to stationary points or minimizers of  $\|c(x)\|^2$ .

## ILL CONDITIONING AND REFORMULATIONS

We now examine the nature of the ill conditioning in the Hessian  $\nabla_{xx}^2 Q(x; \mu_k)$ . An understanding of the properties of this matrix, and the similar Hessians that arise in other penalty and barrier methods, is essential in choosing effective algorithms for the minimization problem and for the linear algebra calculations at each iteration.

The Hessian is given by the formula

$$\nabla_{xx}^2 Q(x; \mu_k) = \nabla^2 f(x) + \sum_{i \in \mathcal{E}} \mu_k c_i(x) \nabla^2 c_i(x) + \mu_k A(x)^T A(x), \quad (17.18)$$

where we have used the definition (17.15) of  $A(x)$ . When  $x$  is close to the minimizer of  $Q(\cdot; \mu_k)$  and the conditions of Theorem 17.2 are satisfied, we have from (17.10) that the sum of the first two terms on the right-hand-side of (17.18) is approximately equal to the Hessian of the Lagrangian function defined in (12.33). To be specific, we have

$$\nabla_{xx}^2 Q(x; \mu_k) \approx \nabla_{xx}^2 \mathcal{L}(x, \lambda^*) + \mu_k A(x)^T A(x), \quad (17.19)$$

when  $x$  is close to the minimizer of  $Q(\cdot; \mu_k)$ . We see from this expression that  $\nabla_{xx}^2 Q(x; \mu_k)$  is approximately equal to the sum of

- a matrix whose elements are independent of  $\mu_k$  (the Lagrangian term), and
- a matrix of rank  $|\mathcal{E}|$  whose nonzero eigenvalues are of order  $\mu_k$  (the second term on the right-hand side of (17.19)).

The number of constraints  $|\mathcal{E}|$  is usually smaller than  $n$ . In this case, the last term in (17.19) is singular. The overall matrix has some of its eigenvalues approaching a constant, while others

are of order  $\mu_k$ . Since  $\mu_k$  is approaching  $\infty$ , the increasing ill conditioning of  $\nabla_{xx}^2 Q(x; \mu_k)$  is apparent.

One consequence of the ill conditioning is possible inaccuracy in the calculation of the Newton step for  $Q(x; \mu_k)$ , which is obtained by solving the following system:

$$\nabla_{xx}^2 Q(x; \mu_k) p = -\nabla_x Q(x; \mu_k). \quad (17.20)$$

In general, the poor conditioning of this system will lead to significant errors in the computed value of  $p$ , regardless of the computational technique used to solve (17.20). For the same reason, iterative methods can be expected to perform poorly unless accompanied by a preconditioning strategy that removes the systematic ill conditioning.

There is an alternative formulation of the equations (17.20) that avoids the ill conditioning due to the final term in (17.18). By introducing a new variable vector  $\zeta$  defined by  $\zeta = \mu A(x)p$ , we see that the vector  $p$  that solves (17.20) also satisfies the following system:

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i \in \mathcal{E}} \mu_k c_i(x) \nabla^2 c_i(x) & A(x)^T \\ A(x) & -(1/\mu_k)I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x; \mu_k) \\ 0 \end{bmatrix}. \quad (17.21)$$

When  $x$  is not too far from the solution  $x^*$ , the coefficient matrix in this system does not have large singular values (of order  $\mu_k$ ), so the system (17.21) can be viewed as a well conditioned reformulation of (17.20). We note, however, that neither system may yield a good search direction  $p$  because the coefficients  $\mu_k c_i(x)$  in the summation term of the upper left block of (17.21) may be poor approximations to the Lagrange multipliers  $-\lambda_i^*$ , even when  $x$  is quite close to the minimizer  $x_k$  of  $Q(x; \mu_k)$ . This fact may cause the quadratic model on which  $p$  is based to be an inadequate model of  $Q(\cdot; \mu_k)$ , so the Newton step may be intrinsically an unsuitable search direction. We discussed possible remedies for this difficulty above, in our comments following Framework 17.1.

To compute the step via (17.21) involves the solution of a linear system of dimension  $n + |\mathcal{E}|$  rather than the system of dimension  $n$  given by (17.19). A similar system must be solved to calculate the sequential quadratic programming (SQP) step (18.6), which is derived in Chapter 18. In fact, when  $\mu_k$  is large, (17.21) can be viewed as a regularization of the SQP step (18.6) in which the term  $-(1/\mu_k)I$  helps to ensure that the iteration matrix is nonsingular even when the Jacobian  $A(x)$  is rank deficient. On the other hand, when  $\mu_k$  is small, (17.21) shows that the step computed by the quadratic penalty method does not closely satisfy the linearization of the constraints. This situation is undesirable because the steps may not make significant progress toward the feasible region, resulting in inefficient global behavior. Moreover, if  $\{\mu_k\}$  does not approach  $\infty$  rapidly enough, we lose the possibility of a superlinear rate that occurs when the linearization is exact; see Chapter 18.

To conclude, the formulation (17.21) allows us to view the quadratic penalty method either as the application of unconstrained minimization to the penalty function  $Q(\cdot; \mu_k)$  or as a variation on the SQP methods discussed in Chapter 18.

## 17.2 NONSMOOTH PENALTY FUNCTIONS

Some penalty functions are *exact*, which means that, for certain choices of their penalty parameters, a single minimization with respect to  $x$  can yield the exact solution of the nonlinear programming problem. This property is desirable because it makes the performance of penalty methods less dependent on the strategy for updating the penalty parameter. The quadratic penalty function of Section 17.1 is not exact because its minimizer is generally not the same as the solution of the nonlinear program for any positive value of  $\mu$ . In this section we discuss *nonsmooth exact* penalty functions, which have proved to be useful in a number of practical contexts.

A popular nonsmooth penalty function for the general nonlinear programming problem (17.6) is the  $\ell_1$  *penalty function* defined by

$$\phi_1(x; \mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} [c_i(x)]^-, \quad (17.22)$$

where we use again the notation  $[y]^- = \max\{0, -y\}$ . Its name derives from the fact that the penalty term is  $\mu$  times the  $\ell_1$  norm of the constraint violation. Note that  $\phi_1(x; \mu)$  is not differentiable at some  $x$ , because of the presence of the absolute value and  $[\cdot]^-$  functions.

The following result establishes the *exactness* of the  $\ell_1$  penalty function. For a proof see [165, Theorem 4.4].

### Theorem 17.3.

Suppose that  $x^*$  is a strict local solution of the nonlinear programming problem (17.6) at which the first-order necessary conditions of Theorem 12.1 are satisfied, with Lagrange multipliers  $\lambda_i^*$ ,  $i \in \mathcal{E} \cup \mathcal{I}$ . Then  $x^*$  is a local minimizer of  $\phi_1(x; \mu)$  for all  $\mu > \mu^*$ , where

$$\mu^* = \|\lambda^*\|_\infty = \max_{i \in \mathcal{E} \cup \mathcal{I}} |\lambda_i^*|. \quad (17.23)$$

If, in addition, the second-order sufficient conditions of Theorem 12.6 hold and  $\mu > \mu^*$ , then  $x^*$  is a strict local minimizer of  $\phi_1(x; \mu)$ .

Loosely speaking, at a solution of the nonlinear program  $x^*$ , any move into the infeasible region is penalized sharply enough that it produces an increase in the penalty function to a value greater than  $\phi_1(x^*; \mu) = f(x^*)$ , thereby forcing the minimizer of  $\phi_1(\cdot; \mu)$  to lie at  $x^*$ .

---

**□ EXAMPLE 17.2**

Consider the following problem in one variable:

$$\min x \quad \text{subject to} \quad x \geq 1, \quad (17.24)$$

whose solution is  $x^* = 1$ . We have that

$$\phi_1(x; \mu) = x + \mu[x - 1]^- = \begin{cases} (1 - \mu)x + \mu & \text{if } x \leq 1, \\ x & \text{if } x > 1. \end{cases} \quad (17.25)$$

As can be seen in Figure 17.3, the penalty function has a minimizer at  $x^* = 1$  when  $\mu > 1$ , but is a monotone increasing function when  $\mu < 1$ . □

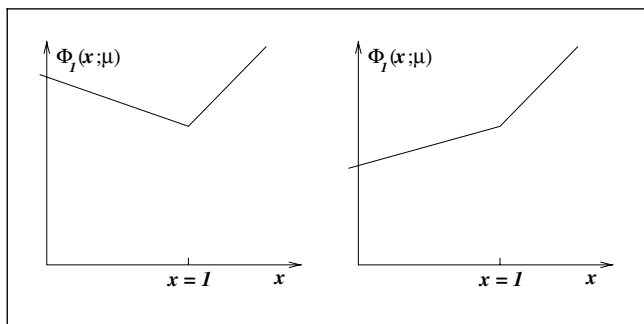
---

Since penalty methods work by minimizing the penalty function directly, we need to characterize stationary points of  $\phi_1$ . Even though  $\phi_1$  is not differentiable, it has a directional derivative  $D(\phi_1(x; \mu); p)$  along any direction; see (A.51) and the example following this definition.

**Definition 17.1.**

A point  $\hat{x} \in R^n$  is a stationary point for the penalty function  $\phi_1(x; \mu)$  if

$$D(\phi_1(\hat{x}; \mu); p) \geq 0, \quad (17.26)$$



**Figure 17.3** Penalty function for problem (17.24) with  $\mu > 1$  (left) and  $\mu < 1$  (right).

for all  $p \in \mathbb{R}^n$ . Similarly,  $\hat{x}$  is a stationary point of the measure of infeasibility

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^- \quad (17.27)$$

if  $D(h(\hat{x}); p) \geq 0$  for all  $p \in \mathbb{R}^n$ . If a point is infeasible for (17.6) but stationary with respect to the infeasibility measure  $h$ , we say that it is an infeasible stationary point.

For the function in Example 17.2, we have for  $x^* = 1$  that

$$D(\phi_1(x^*; \mu); p) = \begin{cases} p & \text{if } p \geq 0 \\ (1 - \mu)p & \text{if } p < 0; \end{cases}$$

it follows that when  $\mu > 1$ , we have  $D(\phi_1(x^*; \mu); p) \geq 0$  for all  $p \in \mathbb{R}$ .

The following result complements Theorem 17.3 by showing that stationary points of  $\phi_1(x; \mu)$  correspond to KKT points of the constrained optimization problem (17.6) under certain assumptions.

#### Theorem 17.4.

Suppose that  $\hat{x}$  is a stationary point of the penalty function  $\phi_1(x; \mu)$  for all  $\mu$  greater than a certain threshold  $\hat{\mu} > 0$ . Then, if  $\hat{x}$  is feasible for the nonlinear program (17.6), it satisfies the KKT conditions (12.34) for (17.6). If  $\hat{x}$  is not feasible for (17.6), it is an infeasible stationary point.

PROOF. Suppose first that  $\hat{x}$  is feasible. We have from (A.51) and the definition (17.22) of  $\phi_1$  that

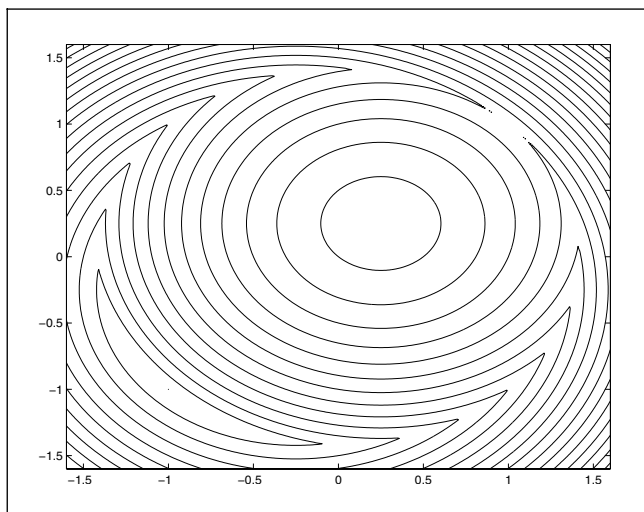
$$D(\phi_1(\hat{x}; \mu); p) = \nabla f(\hat{x})^T p + \mu \sum_{i \in \mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i \in \mathcal{I} \cap \mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- , \quad (17.28)$$

where the active set  $\mathcal{A}(\hat{x})$  is defined in Definition 12.1. (We leave verification of (17.28) as an exercise.) Consider any direction  $p$  in the linearized feasible direction set  $\mathcal{F}(\hat{x})$  of Definition 12.3. By the properties of  $\mathcal{F}(\hat{x})$ , we have

$$|\nabla c_i(\hat{x})^T p| + \sum_{i \in \mathcal{I} \cap \mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- = 0,$$

so that by the stationarity assumption on  $\phi_1(\hat{x}; \mu)$ , we have

$$0 \leq D(\phi_1(\hat{x}; \mu); p) = \nabla f(\hat{x})^T p, \quad \text{for all } p \in \mathcal{F}(\hat{x}).$$



**Figure 17.4** Contours of  $\phi_1(x; \mu)$  from (17.3) for  $\mu = 2$ , contour spacing 0.5.

We can now apply Farkas' Lemma (Lemma 12.4) to deduce that

$$\nabla f(\hat{x}) = \sum_{i \in \mathcal{A}(\hat{x})} \hat{\lambda}_i \nabla c_i(\hat{x}),$$

for some coefficients  $\hat{\lambda}_i$  with  $\hat{\lambda}_i \geq 0$  for all  $i \in \mathcal{I} \cap \mathcal{A}(\hat{x})$ . As we noted earlier (see Theorem 12.1 and (12.35)), this expression implies that the KKT conditions (12.34) hold, as claimed.

We leave the second part of the proof (concerning infeasible  $\hat{x}$ ) as an exercise.  $\square$

---

**EXAMPLE 17.3**

Consider again problem (17.3), for which the  $\ell_1$  penalty function is

$$\phi_1(x; \mu) = x_1 + x_2 + \mu |x_1^2 + x_2^2 - 2|. \quad (17.29)$$

Figure 17.4 plots the function  $\phi_1(x; 2)$ , whose minimizer is the solution  $x^* = (-1, -1)^T$  of (17.3). In fact, following Theorem 17.3, we find that for all  $\mu > |\lambda^*| = 0.5$ , the minimizer of  $\phi_1(x; \mu)$  coincides with  $x^*$ . The sharp corners on the contours indicate nonsmoothness along the boundary of the circle defined by  $x_1^2 + x_2^2 = 2$ .  $\square$

---

These results provide the motivation for an algorithmic framework based on the  $\ell_1$  penalty function, which we now present.

**Framework 17.2** (Classical  $\ell_1$  Penalty Method).

Given  $\mu_0 > 0$ , tolerance  $\tau > 0$ , starting point  $x_0^s$ ;

**for**  $k = 0, 1, 2, \dots$

Find an approximate minimizer  $x_k$  of  $\phi_1(x; \mu_k)$ , starting at  $x_k^s$ ;

**if**  $h(x_k) \leq \tau$

**stop** with approximate solution  $x_k$ ;

**end (if)**

Choose new penalty parameter  $\mu_{k+1} > \mu_k$ ;

Choose new starting point  $x_{k+1}^s$ ;

**end (for)**

The minimization of  $\phi_1(x; \mu_k)$  is made difficult by the nonsmoothness of the function. Nevertheless, as we discuss below, it is well understood how to compute minimization steps using a smooth model of  $\phi_1(x; \mu_k)$ , in a way that resembles SQP methods.

The simplest scheme for updating the penalty parameter  $\mu_k$  is to increase it by a constant multiple (say 5 or 10), if the current value produces a minimizer that is not feasible to within the tolerance  $\tau$ . This scheme sometimes works well in practice, but can also be inefficient. If the initial penalty parameter  $\mu_0$  is too small, many cycles of Framework 17.2 may be needed to determine an appropriate value. In addition, the iterates may move away from the solution  $x^*$  in these initial cycles, in which case the minimization of  $\phi_1(x; \mu_k)$  should be terminated early and  $x_k^s$  should possibly be reset to a previous iterate. If, on the other hand,  $\mu_k$  is excessively large, the penalty function will be difficult to minimize, possibly requiring a large number of iterations. We return to the issue of selecting the penalty parameter below.

## A PRACTICAL $\ell_1$ PENALTY METHOD

As noted already,  $\phi_1(x; \mu)$  is nonsmooth—its gradient is not defined at any  $x$  for which  $c_i(x) = 0$  for some  $i \in \mathcal{E} \cup \mathcal{I}$ . Rather than using techniques for nondifferentiable optimization, such as bundle methods [170], we prefer techniques that take account of the special nature of the nondifferentiabilities in this function. As in the algorithms for unconstrained optimization discussed in the first part of this book, we obtain a step toward the minimizer of  $\phi_1(x; \mu)$  by forming a simplified model of this function and seeking the minimizer of this model. Here, the model can be defined by linearizing the constraints  $c_i$  and replacing the nonlinear programming objective  $f$  by a quadratic

function, as follows:

$$\begin{aligned}
 q(p; \mu) &= f(x) + \nabla f(x)^T p + \frac{1}{2} p^T W p + \mu \sum_{i \in \mathcal{E}} |c_i(x) + \nabla c_i(x)^T p| + \\
 &\quad \mu \sum_{i \in \mathcal{I}} [c_i(x) + \nabla c_i(x)^T p]^-,
 \end{aligned} \tag{17.30}$$

where  $W$  is a symmetric matrix which usually contains second derivative information about  $f$  and  $c_i$ ,  $i \in \mathcal{E} \cup \mathcal{I}$ . The model  $q(p; \mu)$  is not smooth, but we can formulate the problem of minimizing  $q$  as a smooth quadratic programming problem by introducing artificial variables  $r_i$ ,  $s_i$ , and  $t_i$ , as follows:

$$\begin{aligned}
 \min_{p, r, s, t} \quad & f(x) + \frac{1}{2} p^T W p + \nabla f(x)^T p + \mu \sum_{i \in \mathcal{E}} (r_i + s_i) + \mu \sum_{i \in \mathcal{I}} t_i \\
 \text{subject to} \quad & \nabla c_i(x)^T p + c_i(x) = r_i - s_i, \quad i \in \mathcal{E} \\
 & \nabla c_i(x)^T p + c_i(x) \geq -t_i, \quad i \in \mathcal{I} \\
 & r, s, t \geq 0.
 \end{aligned} \tag{17.31}$$

This subproblem can be solved with a standard quadratic programming solver. Even after addition of a “box-shaped” trust region constraint of the form  $\|p\|_\infty \leq \Delta$ , it remains a quadratic program. This approach to minimizing  $\phi_1$  is closely related to sequential quadratic programming (SQP) and will be discussed further in Chapter 18.

The strategy for choosing and updating the penalty parameter  $\mu_k$  is crucial to the practical success of the iteration. We mentioned that a simple (but not always effective) approach is to choose an initial value and increase it repeatedly until feasibility is attained. In some variants of the approach, the penalty parameter is chosen at every iteration so that  $\mu_k > \|\lambda_k\|_\infty$ , where  $\lambda_k$  is an estimate of the Lagrange multipliers computed at  $x_k$ . We base this strategy on Theorem 17.2, which suggests that in a neighborhood of a solution  $x^*$ , a good choice would be to set  $\mu_k$  modestly larger than  $\|\lambda^*\|_\infty$ . This strategy is not always successful, as the multiplier estimates may be inaccurate and may in any case not provide a good appropriate value of  $\mu_k$  far from the solution.

The difficulties of choosing appropriate values of  $\mu_k$  caused nonsmooth penalty methods to fall out of favor during the 1990s and stimulated the development of filter methods, which do not require the choice of a penalty parameter (see Section 15.4). In recent years, however, there has been a resurgence of interest in penalty methods, in part because of their ability to handle degenerate problems. New approaches for updating the penalty parameter appear to have largely overcome the difficulties associated with choosing  $\mu_k$ , at least for some particular implementations (see Algorithm 18.5).

Careful consideration should also be given to the choice of starting point  $x_{k+1}^s$  for the minimization of  $\phi_1(x; \mu_{k+1})$ . If the penalty parameter  $\mu_k$  for the present cycle is appropriate, in the sense that the algorithm made progress toward feasibility, then we can set  $x_{k+1}^s$  to be



the minimizer  $x_k$  of  $\phi_1(x; \mu_k)$  obtained on this cycle. Otherwise, we may want to restore the initial point from an earlier cycle.

### A GENERAL CLASS OF NONSMOOTH PENALTY METHODS

Exact nonsmooth penalty functions can be defined in terms of norms other than the  $\ell_1$  norm. We can write

$$\phi(x; \mu) = f(x) + \mu \|c_{\mathcal{E}}(x)\| + \mu \|[c_{\mathcal{I}}(x)]^-\|, \quad (17.32)$$

where  $\|\cdot\|$  is any vector norm, and all the equality and inequality constraints have been grouped in the vector functions  $c_{\mathcal{E}}$  and  $c_{\mathcal{I}}$ , respectively. Framework 17.2 applies to any of these penalty functions; we simply redefine the measure of infeasibility as  $h(x) = \|c_{\mathcal{E}}(x)\| + \|[c_{\mathcal{I}}(x)]^-\|$ . The most common norms used in practice are the  $\ell_1$ ,  $\ell_\infty$  and  $\ell_2$  (not squared). It is easy to find a reformulation similar to (17.31) for the  $\ell_\infty$  norm.

The theoretical properties described for the  $\ell_1$  function extend to the general class (17.32). In Theorem 17.3, we replace the inequality (17.23) by

$$\mu^* = \|\lambda^*\|_D, \quad (17.33)$$

where  $\|\cdot\|_D$  is the dual norm of  $\|\cdot\|$ , defined in (A.6). Theorem 17.4 applies without modification.

We show now that penalty functions of the type considered so far in this chapter *must* be nonsmooth to be exact. For simplicity, we restrict our attention to the case when there is a single equality constraint  $c_1(x) = 0$ , and consider a penalty function of the form

$$\phi(x; \mu) = f(x) + \mu h(c_1(x)), \quad (17.34)$$

where  $h : \mathbb{R} \rightarrow \mathbb{R}$  is a function satisfying the properties  $h(y) \geq 0$  for all  $y \in \mathbb{R}$  and  $h(0) = 0$ . Suppose for contradiction that  $h$  is continuously differentiable. Since  $h$  has a minimizer at zero, we have from Theorem 2.2 that  $\nabla h(0) = 0$ . If  $x^*$  is a local solution of the problem (17.6), we have  $c_1(x^*) = 0$  and therefore  $\nabla h(c_1(x^*)) = 0$ . If  $x^*$  is a local minimizer of  $\phi(x; \mu)$ , we therefore have

$$0 = \nabla \phi(x^*; \mu) = \nabla f(x^*) + \mu \nabla c_1(x^*) \nabla h(c_1(x^*)) = \nabla f(x^*).$$

However, it is not generally true that the gradient of  $f$  vanishes at the solution of a *constrained* optimization problem, so our original assumption that  $h$  is continuously differentiable must be incorrect, and  $\phi(\cdot; \mu)$  cannot be smooth.

Nonsmooth penalty functions are also used as *merit functions* in methods that compute steps by some other mechanism. For further details see the general discussion of Section 15.4 and the concrete implementations given in Chapters 18 and 19.

### 17.3 AUGMENTED LAGRANGIAN METHOD: EQUALITY CONSTRAINTS

We now discuss an approach known as the *method of multipliers* or the *augmented Lagrangian method*. This algorithm is related to the quadratic penalty algorithm of Section 17.1, but it reduces the possibility of ill conditioning by introducing explicit Lagrange multiplier estimates into the function to be minimized, which is known as the augmented Lagrangian function. In contrast to the penalty functions discussed in Section 17.2, the augmented Lagrangian function largely preserves smoothness, and implementations can be constructed from standard software for unconstrained or bound-constrained optimization.

In this section we use superscripts (usually  $k$  and  $k + 1$ ) on the Lagrange multiplier estimates to denote iteration index, and subscripts (usually  $i$ ) to denote the component indices of the vector  $\lambda$ . For all other variables we use subscripts for the iteration index, as usual.

#### MOTIVATION AND ALGORITHMIC FRAMEWORK

We consider first the equality-constrained problem (17.1). The quadratic penalty function  $Q(x; \mu)$  defined by (17.2) penalizes constraint violations by squaring the infeasibilities and scaling them by  $\mu/2$ . As we see from Theorem 17.2, however, the approximate minimizers  $x_k$  of  $Q(x; \mu_k)$  do not quite satisfy the feasibility conditions  $c_i(x) = 0$ ,  $i \in \mathcal{E}$ . Instead, they are perturbed (see (17.10)) so that

$$c_i(x_k) \approx -\lambda_i^*/\mu_k, \quad \text{for all } i \in \mathcal{E}. \quad (17.35)$$

To be sure, we have  $c_i(x_k) \rightarrow 0$  as  $\mu_k \uparrow \infty$ , but one may ask whether we can alter the function  $Q(x; \mu_k)$  to avoid this systematic perturbation—that is, to make the approximate minimizers more nearly satisfy the equality constraints  $c_i(x) = 0$ , even for moderate values of  $\mu_k$ .

The augmented Lagrangian function  $\mathcal{L}_A(x, \lambda; \mu)$  achieves this goal by including an explicit estimate of the Lagrange multipliers  $\lambda$ , based on the estimate (17.35), in the objective. From the definition

$$\mathcal{L}_A(x, \lambda; \mu) \stackrel{\text{def}}{=} f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x), \quad (17.36)$$

we see that the augmented Lagrangian differs from the (standard) Lagrangian (12.33) for (17.1) by the presence of the squared terms, while it differs from the quadratic penalty function (17.2) in the presence of the summation term involving  $\lambda$ . In this sense, it is a combination of the Lagrangian function and the quadratic penalty function.

We now design an algorithm that fixes the penalty parameter  $\mu$  to some value  $\mu_k > 0$  at its  $k$ th iteration (as in Frameworks 17.1 and 17.2), fixes  $\lambda$  at the current estimate  $\lambda^k$ , and

performs minimization with respect to  $x$ . Using  $x_k$  to denote the approximate minimizer of  $\mathcal{L}_A(x, \lambda^k; \mu_k)$ , we have by the optimality conditions for unconstrained minimization (Theorem 2.2) that

$$0 \approx \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k) = \nabla f(x_k) - \sum_{i \in \mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)] \nabla c_i(x_k). \quad (17.37)$$

By comparing with the optimality condition (17.17) for (17.1), we can deduce that

$$\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k), \quad \text{for all } i \in \mathcal{E}. \quad (17.38)$$

By rearranging this expression, we have that

$$c_i(x_k) \approx -\frac{1}{\mu_k} (\lambda_i^* - \lambda_i^k), \quad \text{for all } i \in \mathcal{E},$$

so we conclude that if  $\lambda^k$  is close to the optimal multiplier vector  $\lambda^*$ , the infeasibility in  $x_k$  will be much smaller than  $(1/\mu_k)$ , rather than being proportional to  $(1/\mu_k)$  as in (17.35). The relation (17.38) immediately suggests a formula for improving our current estimate  $\lambda^k$  of the Lagrange multiplier vector, using the approximate minimizer  $x_k$  just calculated: We can set

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \quad \text{for all } i \in \mathcal{E}. \quad (17.39)$$

This discussion motivates the following algorithmic framework.

**Framework 17.3** (Augmented Lagrangian Method-Equality Constraints).

Given  $\mu_0 > 0$ , tolerance  $\tau_0 > 0$ , starting points  $x_0^s$  and  $\lambda^0$ ;

**for**  $k = 0, 1, 2, \dots$

Find an approximate minimizer  $x_k$  of  $\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$ , starting at  $x_k^s$ ,  
and terminating when  $\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \leq \tau_k$ ;

**if** a convergence test for (17.1) is satisfied

**stop** with approximate solution  $x_k$ ;

**end (if)**

Update Lagrange multipliers using (17.39) to obtain  $\lambda^{k+1}$ ;

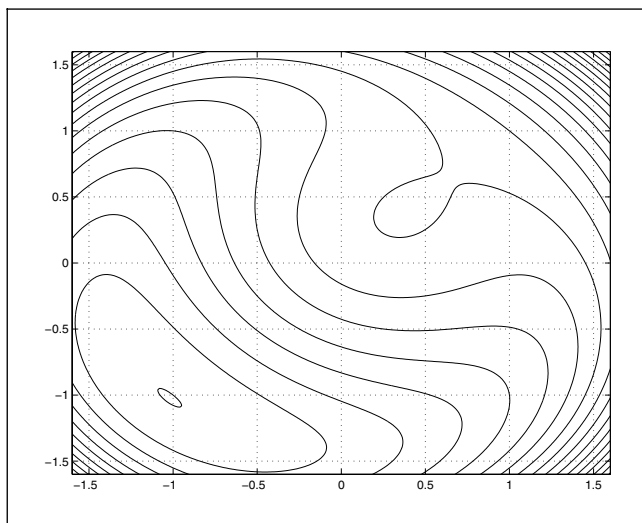
Choose new penalty parameter  $\mu_{k+1} \geq \mu_k$ ;

Set starting point for the next iteration to  $x_{k+1}^s = x_k$ ;

Select tolerance  $\tau_{k+1}$ ;

**end (for)**

We show below that convergence of this method can be assured without increasing  $\mu$  indefinitely. Ill conditioning is therefore less of a problem than in Framework 17.1, so the choice of starting point  $x_{k+1}^s$  in Framework 17.3 is less critical. (In Framework 17.3 we



**Figure 17.5** Contours of  $\mathcal{L}_A(x, \lambda; \mu)$  from (17.40) for  $\lambda = -0.4$  and  $\mu = 1$ , contour spacing 0.5.

simply start the search at iteration  $k + 1$  from the previous approximate minimizer  $x_k$ .) The tolerance  $\tau_k$  could be chosen to depend on the infeasibility  $\sum_{i \in \mathcal{E}} |c(x_k)|$ , and the penalty parameter  $\mu$  may be increased if the reduction in this infeasibility measure is insufficient at the present iteration.

---

□ **EXAMPLE 17.4**

Consider again problem (17.3), for which the augmented Lagrangian is

$$\mathcal{L}_A(x, \lambda; \mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2. \quad (17.40)$$

The solution of (17.3) is  $x^* = (-1, -1)^T$  and the optimal Lagrange multiplier is  $\lambda^* = -0.5$ .

Suppose that at iterate  $k$  we have  $\mu_k = 1$  (as in Figure 17.1), while the current multiplier estimate is  $\lambda^k = -0.4$ . Figure 17.5 plots the function  $\mathcal{L}_A(x, -0.4; 1)$ . Note that the spacing of the contours indicates that the conditioning of this problem is similar to that of the quadratic penalty function  $Q(x; 1)$  illustrated in Figure 17.1. However, the minimizing value of  $x_k \approx (-1.02, -1.02)^T$  is much closer to the solution  $x^* = (-1, -1)^T$  than is the minimizing value of  $Q(x; 1)$ , which is approximately  $(-1.1, -1.1)^T$ . This example shows that the inclusion of the Lagrange multiplier term in the function  $\mathcal{L}_A(x, \lambda; \mu)$  can result in a significant improvement over the quadratic penalty method, as a way to reformulate the constrained optimization problem (17.1). □

---

### PROPERTIES OF THE AUGMENTED LAGRANGIAN

We now prove two results that justify the use of the augmented Lagrangian function and the method of multipliers for equality-constrained problems.

The first result validates the approach of Framework 17.3 by showing that when we have knowledge of the exact Lagrange multiplier vector  $\lambda^*$ , the solution  $x^*$  of (17.1) is a strict minimizer of  $\mathcal{L}_A(x, \lambda^*; \mu)$  for all  $\mu$  sufficiently large. Although we do not know  $\lambda^*$  exactly in practice, the result and its proof suggest that we can obtain a good estimate of  $x^*$  by minimizing  $\mathcal{L}_A(x, \lambda; \mu)$  even when  $\mu$  is not particularly large, provided that  $\lambda$  is a reasonably good estimate of  $\lambda^*$ .

#### Theorem 17.5.

*Let  $x^*$  be a local solution of (17.1) at which the LICQ is satisfied (that is, the gradients  $\nabla c_i(x^*)$ ,  $i \in \mathcal{E}$ , are linearly independent vectors), and the second-order sufficient conditions specified in Theorem 12.6 are satisfied for  $\lambda = \lambda^*$ . Then there is a threshold value  $\bar{\mu}$  such that for all  $\mu \geq \bar{\mu}$ ,  $x^*$  is a strict local minimizer of  $\mathcal{L}_A(x, \lambda^*; \mu)$ .*

PROOF. We prove the result by showing that  $x^*$  satisfies the second-order sufficient conditions to be a strict local minimizer of  $\mathcal{L}_A(x, \lambda^*; \mu)$  (see Theorem 2.4) for all  $\mu$  sufficiently large; that is,

$$\nabla_x \mathcal{L}_A(x^*, \lambda^*; \mu) = 0, \quad \nabla_{xx}^2 \mathcal{L}_A(x^*, \lambda^*; \mu) \text{ positive definite.} \quad (17.41)$$

Because  $x^*$  is a local solution for (17.1) at which LICQ is satisfied, we can apply Theorem 12.1 to deduce that  $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$  and  $c_i(x^*) = 0$  for all  $i \in \mathcal{E}$ , so that

$$\begin{aligned} \nabla_x \mathcal{L}_A(x^*, \lambda^*; \mu) &= \nabla f(x^*) - \sum_{i \in \mathcal{E}} [\lambda_i^* - \mu c_i(x^*)] \nabla c_i(x^*) \\ &= \nabla f(x^*) - \sum_{i \in \mathcal{E}} \lambda_i^* \nabla c_i(x^*) = \nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \end{aligned}$$

verifying the first part of (17.41), independently of  $\mu$ .

For the second part of (17.41), we define  $A$  to be the constraint gradient matrix in (17.15) evaluated at  $x^*$ , and write

$$\nabla_{xx}^2 \mathcal{L}_A(x^*, \lambda^*; \mu) = \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) + \mu A^T A.$$

If the claim in (17.41) were not true, then for each integer  $k \geq 1$ , we could choose a vector  $w_k$  with  $\|w_k\| = 1$  such that

$$0 \geq w_k^T \nabla_{xx}^2 \mathcal{L}_A(x^*, \lambda^*; k) w_k = w_k^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w_k + k \|A w_k\|_2^2, \quad (17.42)$$

and therefore

$$\|Aw_k\|_2^2 \leq -(1/k)w_k^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w_k \rightarrow 0, \quad \text{as } k \rightarrow \infty. \quad (17.43)$$

Since the vectors  $\{w_k\}$  lie in a compact set (the surface of the unit sphere), they have an accumulation point  $w$ . The limit (17.43) implies that  $Aw = 0$ . Moreover, by rearranging (17.42), we have that

$$w_k^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w_k \leq -k \|Aw_k\|_2^2 \leq 0,$$

so by taking limits we have  $w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w \leq 0$ . However, this inequality contradicts the second-order conditions in Theorem 12.6 which, when applied to (17.1), state that we must have  $w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w > 0$  for all nonzero vectors  $w$  with  $Aw = 0$ . Hence, the second part of (17.41) holds for all  $\mu$  sufficiently large.  $\square$

The second result, given by Bertsekas [19, Proposition 4.2.3], describes the more realistic situation of  $\lambda \neq \lambda^*$ . It gives conditions under which there is a minimizer of  $\mathcal{L}_A(x, \lambda; \mu)$  that lies close to  $x^*$  and gives error bounds on both  $x_k$  and the updated multiplier estimate  $\lambda^{k+1}$  obtained from solving the subproblem at iteration  $k$ .

**Theorem 17.6.**

*Suppose that the assumptions of Theorem 17.5 are satisfied at  $x^*$  and  $\lambda^*$  and let  $\bar{\mu}$  be chosen as in that theorem. Then there exist positive scalars  $\delta, \epsilon$ , and  $M$  such that the following claims hold:*

(a) *For all  $\lambda^k$  and  $\mu_k$  satisfying*

$$\|\lambda^k - \lambda^*\| \leq \mu_k \delta, \quad \mu_k \geq \bar{\mu}, \quad (17.44)$$

*the problem*

$$\min_x \mathcal{L}_A(x, \lambda^k; \mu_k) \quad \text{subject to } \|x - x^*\| \leq \epsilon$$

*has a unique solution  $x_k$ . Moreover, we have*

$$\|x_k - x^*\| \leq M \|\lambda^k - \lambda^*\| / \mu_k. \quad (17.45)$$

(b) *For all  $\lambda^k$  and  $\mu_k$  that satisfy (17.44), we have*

$$\|\lambda^{k+1} - \lambda^*\| \leq M \|\lambda^k - \lambda^*\| / \mu_k, \quad (17.46)$$

*where  $\lambda^{k+1}$  is given by the formula (17.39).*

(c) For all  $\lambda^k$  and  $\mu_k$  that satisfy (17.44), the matrix  $\nabla_{xx}^2 \mathcal{L}_A(x_k, \lambda^k; \mu_k)$  is positive definite and the constraint gradients  $\nabla c_i(x_k)$ ,  $i \in \mathcal{E}$ , are linearly independent.

This theorem illustrates some salient properties of the augmented Lagrangian approach. The bound (17.45) shows that  $x_k$  will be close to  $x^*$  if  $\lambda_k$  is accurate or if the penalty parameter  $\mu_k$  is large. Hence, this approach gives us two ways of improving the accuracy of  $x_k$ , whereas the quadratic penalty approach gives us only one option: increasing  $\mu_k$ . The bound (17.46) states that, locally, we can ensure an improvement in the accuracy of the multipliers by choosing a sufficiently large value of  $\mu_k$ . The final observation of the theorem shows that second-order sufficient conditions for unconstrained minimization (see Theorem 2.4) are also satisfied for the  $k$ th subproblem under the given conditions, so one can expect good performance by applying standard unconstrained minimization techniques.

## 17.4 PRACTICAL AUGMENTED LAGRANGIAN METHODS

In this section we discuss practical augmented Lagrangian procedures, in particular, procedures for handling inequality constraints. We discuss three approaches based, respectively, on bound-constrained, linearly constrained, and unconstrained formulations. The first two are the basis of the successful nonlinear programming codes LANCELOT [72] and MINOS [218].

### BOUND-CONSTRAINED FORMULATION

Given the general nonlinear program (17.6), we can convert it to a problem with equality constraints and bound constraints by introducing slack variables  $s_i$  and replacing the general inequalities  $c_i(x) \geq 0$ ,  $i \in \mathcal{I}$ , by

$$c_i(x) - s_i = 0, \quad s_i \geq 0, \quad \text{for all } i \in \mathcal{I}. \quad (17.47)$$

Bound constraints,  $l \leq x \leq u$ , need not be transformed. By reformulating in this way, we can write the nonlinear program as follows:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad c_i(x) = 0, \quad i = 1, 2, \dots, m, \quad l \leq x \leq u. \quad (17.48)$$

(The slacks  $s_i$  have been incorporated into the vector  $x$  and the constraint functions  $c_i$  have been redefined accordingly. We have numbered the constraints consecutively with  $i = 1, 2, \dots, m$  and in the discussion below we gather them into the vector function  $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ .) Some of the components of the lower bound vector  $l$  may be set to  $-\infty$ , signifying that there is no lower bound on the components of  $x$  in question; similarly for  $u$ .

The bound-constrained Lagrangian (BCL) approach incorporates only the equality constraints from (17.48) into the augmented Lagrangian, that is,

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^m \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^m c_i^2(x). \quad (17.49)$$

The bound constraints are enforced explicitly in the subproblem, which has the form

$$\min_x \mathcal{L}_A(x, \lambda; \mu) \quad \text{subject to } l \leq x \leq u. \quad (17.50)$$

After this problem has been solved approximately, the multipliers  $\lambda$  and the penalty parameter  $\mu$  are updated and the process is repeated.

An efficient technique for solving the nonlinear program with bound constraints (17.50) (for fixed  $\mu$  and  $\lambda$ ) is the (nonlinear) gradient projection method discussed in Section 18.6. By specializing the KKT conditions (12.34) to the problem (17.50), we find that the first-order necessary condition for  $x$  to be a solution of (17.50) is that

$$x - P(x - \nabla_x \mathcal{L}_A(x, \lambda; \mu), l, u) = 0, \quad (17.51)$$

where  $P(g, l, u)$  is the projection of the vector  $g \in \mathbb{R}^n$  onto the rectangular box  $[l, u]$  defined as follows

$$P(g, l, u)_i = \begin{cases} l_i & \text{if } g_i \leq l_i, \\ g_i & \text{if } g_i \in (l_i, u_i), \\ u_i & \text{if } g_i \geq u_i, \end{cases} \quad \text{for all } i = 1, 2, \dots, n. \quad (17.52)$$

We are now ready to describe the algorithm implemented in the LANCELOT software package.

**Algorithm 17.4** (Bound-Constrained Lagrangian Method).

Choose an initial point  $x_0$  and initial multipliers  $\lambda^0$ ;

Choose convergence tolerances  $\eta_*$  and  $\omega_*$ ;

Set  $\mu_0 = 10$ ,  $\omega_0 = 1/\mu_0$ , and  $\eta_0 = 1/\mu_0^{0.1}$ ;

**for**  $k = 0, 1, 2, \dots$

Find an approximate solution  $x_k$  of the subproblem (17.50) such that

$$\|x_k - P(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u)\| \leq \omega_k;$$

**if**  $\|c(x_k)\| \leq \eta_k$

(\* test for convergence \*)

**if**  $\|c(x_k)\| \leq \eta_*$  and  $\|x_k - P(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u)\| \leq \omega_*$

**stop** with approximate solution  $x_k$ ;



```

end (if)
(* update multipliers, tighten tolerances *)
 $\lambda^{k+1} = \lambda^k - \mu_k c(x_k);$ 
 $\mu_{k+1} = \mu_k;$ 
 $\eta_{k+1} = \eta_k / \mu_{k+1}^{0.9};$ 
 $\omega_{k+1} = \omega_k / \mu_{k+1};$ 
else
(* increase penalty parameter, tighten tolerances *)
 $\lambda^{k+1} = \lambda^k;$ 
 $\mu_{k+1} = 100\mu_k;$ 
 $\eta_{k+1} = 1 / \mu_{k+1}^{0.1};$ 
 $\omega_{k+1} = 1 / \mu_{k+1};$ 
end (if)
end (for)

```

The main branch in the algorithm occurs after problem (17.50) has been solved approximately, when the algorithm tests to see if the constraints have decreased sufficiently, as measured by the condition

$$\|c(x_k)\| \leq \eta_k. \quad (17.53)$$

If this condition holds, the penalty parameter is not changed for the next iteration because the current value of  $\mu_k$  is producing an acceptable level of constraint violation. The Lagrange multiplier estimates are updated according to the formula (17.39) and the tolerances  $\omega_k$  and  $\eta_k$  are tightened in advance of the next iteration. If, on the other hand, (17.53) does not hold, then we increase the penalty parameter to ensure that the next subproblem will place more emphasis on decreasing the constraint violations. The Lagrange multiplier estimates are not updated in this case; the focus is on improving feasibility.

The constants 0.1, 0.9, and 100 appearing in Algorithm 17.4 are to some extent arbitrary; other values can be used without compromising theoretical convergence properties. LANCELOT uses the gradient projection method with trust regions (see (18.61)) to solve the bound-constrained nonlinear subproblem (17.50). In this context, the gradient projection method constructs a quadratic model of the augmented Lagrangian  $\mathcal{L}_A$  and computes a step  $d$  by approximately solving the trust region problem

$$\begin{aligned} \min_d \quad & \frac{1}{2}d^T [\nabla_{xx}^2 \mathcal{L}(x_k, \lambda^k) + \mu_k A_k^T A_k] d + \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)^T d \\ \text{subject to} \quad & l \leq x_k + d \leq u, \quad \|d\|_\infty \leq \Delta, \end{aligned} \quad (17.54)$$

where  $A_k = A(x_k)$  and  $\Delta$  is a trust region radius. (We can formulate the trust-region constraint by means of the bounds  $-\Delta e \leq d \leq \Delta e$ , where  $e = (1, 1, \dots, 1)^T$ .) Each iteration of the algorithm for solving this subproblem proceeds in two stages. First, a

projected gradient line search is performed to determine which components of  $d$  should be set at one of their bounds. Second, a conjugate gradient iteration minimizes (17.54) with respect to the free components of  $d$ —those not at one of their bounds. Importantly, this algorithm does not require the factorizations of a KKT matrix or of the constraint Jacobian  $A_k$ . The conjugate gradient iteration only requires matrix-vector products, a feature that makes LANCELOT suitable for large problems.

The Hessian of the Lagrangian  $\nabla_{xx}^2 \mathcal{L}(x_k, \lambda^k)$  in (17.54) can be replaced by a quasi-Newton approximation based on the BFGS or SR1 updating formulas. LANCELOT is designed to take advantage of partially separable structure in the objective function and constraints, either in the evaluation of the Hessian of the Lagrangian or in the quasi-Newton updates (see Section 7.4).

### LINEARLY CONSTRAINED FORMULATION

The principal idea behind *linearly constrained Lagrangian* (LCL) methods is to generate a step by minimizing the Lagrangian (or augmented Lagrangian) subject to linearizations of the constraints. If we use the formulation (17.48) of the nonlinear programming problem, the subproblem used in the LCL approach takes the form

$$\min_x F_k(x) \tag{17.55a}$$

$$\text{subject to } c(x_k) + A_k(x - x_k) = 0, \quad l \leq x \leq u. \tag{17.55b}$$

There are several possible choices for  $F_k(x)$ . Early LCL methods defined

$$F_k(x) = f(x) - \sum_{i=1}^m \lambda_i^k \bar{c}_i^k(x), \tag{17.56}$$

where  $\lambda^k$  is the current Lagrange multiplier estimate and  $\bar{c}_i^k(x)$  is the difference between  $c_i(x)$  and its linearization at  $x_k$ , that is,

$$\bar{c}_i^k(x) = c_i(x) - c_i(x_k) - \nabla c_i(x_k)^T (x - x_k). \tag{17.57}$$

One can show that as  $x_k$  converges to a solution  $x^*$ , the Lagrange multiplier associated with the equality constraint in (17.55b) converges to the optimal multiplier. Therefore, one can set  $\lambda^k$  in (17.56) to be the Lagrange multiplier for the equality constraint in (17.55b) from the previous iteration.

Current LCL methods define  $F_k$  to be the augmented Lagrangian function

$$F_k(x) = f(x) - \sum_{i=1}^m \lambda_i^k \bar{c}_i^k(x) + \frac{\mu}{2} \sum_{i=1}^m [\bar{c}_i^k(x)]^2. \tag{17.58}$$

This definition of  $F_k$  appears to yield more reliable convergence from remote starting points than does (17.56), in practice.

There is a notable similarity between (17.58) and the augmented Lagrangian (17.36), the difference being that the original constraints  $c_i(x)$  have been replaced by the functions  $\bar{c}_i^k(x)$ , which capture only the “second-order and above” terms of  $c_i$ . The subproblem (17.55) differs from the augmented Lagrangian subproblem in that the new  $x$  is required to satisfy exactly a linearization of the equality constraints, while the linear part of each constraint is factored out of the objective via the use of  $\bar{c}_i^k$  in place of  $c_i$ . A procedure similar to the one in Algorithm 17.4 can be used for updating the penalty parameter  $\mu$  and for adjusting the tolerances that govern the accuracy of the solution of the subproblem.

Since  $\bar{c}_i^k(x)$  has zero gradient at  $x = x_k$ , we have that  $\nabla F_k(x_k) = \nabla f(x_k)$ , where  $F_k$  is defined by either (17.56) or (17.58). We can also show that the Hessian of  $F_k$  is closely related to the Hessians of the Lagrangian or augmented Lagrangian functions for (17.1). Because of these properties, the subproblem (17.55) is similar to the SQP subproblems described in Chapter 18, with the quadratic objective in SQP being replaced by a nonlinear objective in LCL.

The well known code MINOS [218] uses the nonlinear model function (17.58) and solves the subproblem via a reduced gradient method that employs quasi-Newton approximations to the reduced Hessian of  $F_k$ . A fairly accurate solution of the subproblem is computed in MINOS to try to ensure that the Lagrange multiplier estimates for the equality constraint in (17.55b) (subsequently used in (17.58)) are of good quality. As a result, MINOS typically requires more evaluations of the objective  $f$  and constraint functions  $c_i$  (and their gradients) in total than SQP methods or interior-point methods. The total number of subproblems (17.55) that are solved in the course of the algorithm is, however, sometimes smaller than in other approaches.

## UNCONSTRAINED FORMULATION

We can obtain an unconstrained form of the augmented Lagrangian subproblem for inequality-constrained problems by using a derivation based on the proximal point approach. Supposing for simplicity that the problem has no equality constraints ( $\mathcal{E} = \emptyset$ ), we can write the problem (17.6) equivalently as an unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} F(x), \quad (17.59)$$

where

$$F(x) = \max_{\lambda \geq 0} \left\{ f(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x) \right\} = \begin{cases} f(x) & \text{if } x \text{ is feasible,} \\ \infty & \text{otherwise.} \end{cases} \quad (17.60)$$

To verify these expressions for  $F$ , consider first the case of  $x$  infeasible, that is,  $c_i(x) < 0$  for some  $i$ . We can then choose  $\lambda_i$  arbitrarily large and positive while setting  $\lambda_j = 0$  for all

$j \neq i$ , to verify that  $F(x)$  is infinite in this case. If  $x$  is feasible, we have  $c_i(x) \geq 0$  for all  $i \in \mathcal{I}$ , so the maximum is attained at  $\lambda = 0$ , and  $F(x) = f(x)$  in this case. By combining (17.59) with (17.60), we have

$$\min_{x \in \mathbb{R}^n} F(x) = \min_{x \text{ feasible}} f(x), \quad (17.61)$$

which is simply the original inequality-constrained problem. It is not practical to minimize  $F$  directly, however, since this function is not smooth—it jumps from a finite value to an infinite value as  $x$  crosses the boundary of the feasible set.

We can make this approach more practical by replacing  $F$  by a smooth approximation  $\hat{F}(x; \lambda^k, \mu_k)$  which depends on the penalty parameter  $\mu_k$  and Lagrange multiplier estimate  $\lambda^k$ . This approximation is defined as follows:

$$\hat{F}(x; \lambda^k, \mu_k) = \max_{\lambda \geq 0} \left\{ f(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x) - \frac{1}{2\mu_k} \sum_{i \in \mathcal{I}} (\lambda_i - \lambda_i^k)^2 \right\}. \quad (17.62)$$

The final term in this expression applies a penalty for any move of  $\lambda$  away from the previous estimate  $\lambda^k$ ; it encourages the new maximizer  $\lambda$  to stay *proximal* to the previous estimate  $\lambda^k$ . Since (17.62) represents a bound-constrained quadratic problem in  $\lambda$ , separable in the individual components  $\lambda_i$ , we can perform the maximization explicitly, to obtain

$$\lambda_i = \begin{cases} 0 & \text{if } -c_i(x) + \lambda_i^k / \mu_k \leq 0; \\ \lambda_i^k - \mu_k c_i(x) & \text{otherwise.} \end{cases} \quad (17.63)$$

By substituting these values in (17.62), we find that

$$\hat{F}(x; \lambda^k, \mu_k) = f(x) + \sum_{i \in \mathcal{I}} \psi(c_i(x), \lambda_i^k; \mu_k), \quad (17.64)$$

where the function  $\psi$  of three scalar arguments is defined as follows:

$$\psi(t, \sigma; \mu) \stackrel{\text{def}}{=} \begin{cases} -\sigma t + \frac{\mu}{2} t^2 & \text{if } t - \sigma / \mu \leq 0, \\ -\frac{1}{2\mu} \sigma^2 & \text{otherwise,} \end{cases} \quad (17.65)$$

Hence, we can obtain the new iterate  $x_k$  by minimizing  $\hat{F}(x; \lambda^k, \mu_k)$  with respect to  $x$ , and use the formula (17.63) to obtain the updated Lagrange multiplier estimates  $\lambda^{k+1}$ . By comparing with Framework 17.3, we see that  $F$  plays the role of  $\mathcal{L}_A$  and that the scheme just described extends the augmented Lagrangian methods for equality constraints neatly to the inequality-constrained case. Unlike the bound-constrained and linearly constrained formulations, however, this unconstrained formulation is not the basis of any widely used software packages, so its practical properties have not been tested.

## 17.5 PERSPECTIVES AND SOFTWARE

The quadratic penalty approach is often used by practitioners when the number of constraints is small. In fact, minimization of  $Q(x; \mu)$  is sometimes performed for just one large value of  $\mu$ . Unless  $\mu$  is chosen wisely (with the benefit of experience with the underlying application), the resulting solution may not be very accurate. Since the main software packages for constrained optimization do not implement a quadratic penalty approach, little attention has been paid to techniques for updating the penalty parameter, adjusting the tolerances  $\tau_k$ , and choosing the starting points  $x_k^s$  for each iteration. (See Gould [141] for a discussion of these issues.)

Despite the intuitive appeal and simplicity of the quadratic penalty method of Framework 17.1, the augmented Lagrangian method of Sections 17.3 and 17.4 is generally preferred. The subproblems are in general no more difficult to solve, and the introduction of multiplier estimates reduces the likelihood that large values of  $\mu$  will be needed to obtain good feasibility and accuracy, thereby avoiding ill conditioning of the subproblem. The quadratic penalty approach remains, however, an important mechanism for regularizing other algorithms such as sequential quadratic programming (SQP) methods, as we mention at the end of Section 17.1.

A general-purpose  $\ell_1$  penalty method was developed by Fletcher in the 1980's. It is known as the  $S\ell_1$ QP method because it has features in common with SQP methods. More recently, an  $\ell_1$  penalty method that uses linear programming subproblems has been implemented as part of the KNITRO [46] software package. These two methods are discussed in Section 18.5.

The  $\ell_1$  penalty function has received significant attention in recent years. It has been successfully used to treat difficult problems, such as mathematical programs with complementarity constraints (MPCCs), in which the constraints do not satisfy standard constraint qualifications [274]. By including these problematic constraints as a penalty term, rather than linearizing them exactly, and treating the remaining constraints using other techniques such as SQP or interior-point, it is possible to extend the range of applicability of these other approaches. See [8] for an active-set method and [16, 191] for interior-point methods for MPCCs. The SNOPT software package uses an  $\ell_1$  penalty approach within an SQP method as a safeguard strategy in case the quadratic model appears to be infeasible or unbounded or to have unbounded multipliers.

Augmented Lagrangian methods have been popular for many years because, in part, of their simplicity. The MINOS and LANCELOT packages rank among the best implementations of augmented Lagrangian methods. Both are suitable for large-scale nonlinear programming problems. At a general level, the linearly constrained Lagrangian (LCL) of MINOS and the bound-constrained Lagrangian (BCL) method of LANCELOT have important features in common. They differ significantly, however, in the formulation of the step-computation subproblems and in the techniques used to solve these subproblems. MINOS follows a reduced-space approach to handle linearized constraints and employs a (dense) quasi-Newton approximation to the Hessian of the Lagrangian. As a result, MINOS

is most successful for problems with relatively few degrees of freedom. LANCELOT, on the other hand, is more effective when there are relatively few constraints. As indicated in Section 17.4, LANCELOT does not require a factorization of the constraint Jacobian matrix  $A$ , again enhancing its suitability for very large problems, and provides a variety of Hessian approximation options and preconditioners. The PENNON software package [184] is based on an augmented Lagrangian approach and has the advantage of permitting semi-definite matrix constraints.

A weakness of both the bound-constrained and unconstrained Lagrangian methods is that they complicate constraints by squaring them in (17.49); progress in feasibility is only achieved through the minimization of the augmented Lagrangian. In contrast, the LCL formulation (17.55) promotes steady progress toward feasibility by performing a Newton-like step on the constraints. Not surprisingly, numerical experience has shown an advantage of MINOS over LANCELOT for problems with linear constraints.

*Smooth* exact penalty functions have been constructed from the augmented Lagrangian functions of Section 17.3, but these are considerably more complicated. As an example, we mention the function of Fletcher for equality-constrained problems, defined as follows:

$$\phi_F(x; \mu) = f(x) - \lambda(x)^T c(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i(x)^2. \quad (17.66)$$

The Lagrange multiplier estimates  $\lambda(x)$  are defined explicitly in terms of  $x$  via the least-squares estimate, defined as

$$\lambda(x) = [A(x)A(x)^T]^{-1}A(x)\nabla f(x). \quad (17.67)$$

The function  $\phi_F$  is differentiable and exact, though the threshold value  $\mu^*$  defining the exactness property is not as easy to specify as for the nonsmooth  $\ell_1$  penalty function. Drawbacks of the penalty function  $\phi_F$  include the cost of evaluating  $\lambda(x)$  via (17.67), the fact that  $\lambda(x)$  is not uniquely defined when  $A(x)$  does not have full rank, and the observation that estimates of  $\lambda$  may be poor when  $A(x)$  is nearly singular.

## NOTES AND REFERENCES

The quadratic penalty function was first proposed by Courant [81]. Gould [140] addresses the issue of stable determination of the Newton step for  $Q(x; \mu_k)$ . His formula (2.2) differs from our formula (17.20) in the right-hand-side, but both systems give rise to the same  $p$  component.

The augmented Lagrangian method was proposed by Hestenes [167] and Powell [240]. In the early days it was known as the “method of multipliers.” A key reference in this area is Bertsekas [18]. Chapters 1–3 of that book contain a thorough motivation of the method that outlines its connections to other approaches. Other introductory discussions

are given by Fletcher [101, Section 12.2], and Polak [236, Section 2.8]. The extension to inequality constraints in the unconstrained formulation was described by Rockafellar [269] and Powell [243].


Linearly constrained Lagrangian methods were proposed by Robinson [266] and Rosen and Kreuser [271]. The MINOS implementation is due to Murtagh and Saunders [218] and the LANCELOT implementation due to Conn, Gould and Toint [72]. We have followed Friedlander and Saunders [114] in our use of the terms “linearly constrained Lagrangian” and “bound-constrained Lagrangian.”


---


 **EXERCISES**


 **17.1**


- (a) Write an equality-constrained problem which has a local solution and for which the quadratic penalty function  $Q$  is unbounded for *any* value of the penalty parameter.
- (b) Write a problem with a single inequality constraint that has the same unboundedness property.

 **17.2** Draw the contour lines of the quadratic penalty function  $Q$  for problem (17.5) corresponding to  $\mu = 1$ . Find the stationary points of  $Q$ .


 **17.3** Minimize the quadratic penalty function for problem (17.3) for  $\mu_k = 1, 10, 100, 1000$  using an unconstrained minimization algorithm. Set  $\tau_k = 1/\mu_k$  in Framework 17.1, and choose the starting point  $x_{k+1}^s$  for each minimization to be the solution for the previous value of the penalty parameter. Report the approximate solution of each penalty function.

 **17.4** For  $z \in \mathbb{R}$ , show that the function  $\min(0, z)^2$  has a discontinuous second derivative at  $z = 0$ . (It follows that quadratic penalty function (17.7) may not have continuous second derivatives even when  $f$  and  $c_i, i \in \mathcal{E} \cup \mathcal{I}$ , in (17.6) are all twice continuously differentiable.)


 **17.5** Write a quadratic program similar to (17.31) for the case when the norm in (17.32) is the infinity norm.

 **17.6** Suppose that a nonlinear program has a minimizer  $x^*$  with Lagrange multiplier vector  $\lambda^*$ . One can show (Fletcher [101, Theorem 14.3.2]) that the function  $\phi_1(x; \mu)$  does not have a local minimizer at  $x^*$  unless  $\mu > \|\lambda^*\|_\infty$ . Verify that this observation holds for Example 17.1.

 **17.7** Verify (17.28).

 **17.8** Prove the second part of Theorem 17.4. That is, if  $\hat{x}$  is a stationary point of  $\phi_1(x; \mu)$  for all  $\mu$  sufficiently large, but  $\hat{x}$  is infeasible for problem (17.6), then  $\hat{x}$  is

an infeasible stationary point. (Hint: Use the fact that  $D(\phi_1(\hat{x}; \mu); p) = \nabla f(\hat{x})^T p + \mu D(h(\hat{x}); p)$ , where  $h$  is defined in (17.27).)


 **17.9** Verify that the KKT conditions for the bound-constrained problem


$$\min_{x \in \mathbb{R}^n} \phi(x) \quad \text{subject to } l \leq x \leq u$$


are equivalent to the compactly stated condition

$$x - P(x - \nabla\phi(x), l, u) = 0,$$

where the projection operator  $P$  onto the rectangular box  $[l, u]$  is defined in (17.52).

 **17.10** Calculate the gradient and Hessian of the LCL objective functions  $F_k(x)$  defined by (17.56) and (17.58). Evaluate these quantities at  $x = x_k$ .

 **17.11** Show that the function  $\psi(t, \sigma; \mu)$  defined in (17.65) has a discontinuity in its second derivative with respect to  $t$  when  $t = \sigma/\mu$ . Assuming that  $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is twice continuously differentiable, write down the second partial derivative matrix of  $\psi(c_i(x), \lambda_i; \mu)$  with respect to  $x$  for the two cases  $c_i(x) < \lambda_i/\mu$  and  $c_i(x) \geq a\lambda_i/\mu$ .

 **17.12** Verify that the multipliers  $\lambda_i$ ,  $i \in \mathcal{I}$  defined in (17.63) are indeed those that attain the maximum in (17.62), and that the equality (17.64) holds. Hint: Use the fact that KKT conditions for the problem

$$\max \phi(x) \quad \text{subject to } x \geq 0$$

indicate that at a stationary point, we either have  $x_i = 0$  and  $[\nabla\phi(x)]_i \leq 0$ , or  $x_i > 0$  and  $[\nabla\phi(x)]_i = 0$ .