Chapter 19

Interior-Point Methods for Nonlinear Programming

Interior-point (or barrier) methods have proved to be as successful for nonlinear optimization as for linear programming, and together with active-set SQP methods, they are currently considered the most powerful algorithms for large-scale nonlinear programming. Some of the key ideas, such as primal-dual steps, carry over directly from the linear programming case, but several important new challenges arise. These include the treatment of nonconvexity, the strategy for updating the barrier parameter in the presence of nonlinearities, and the need to ensure progress toward the solution. In this chapter we describe two classes of interior-point methods that have proved effective in practice.

The methods in the first class can be viewed as direct extensions of interior-point methods for linear and quadratic programming. They use line searches to enforce convergence and employ direct linear algebra (that is, matrix factorizations) to compute steps. The methods in the second class use a quadratic model to define the step and incorporate a trust-region constraint to provide stability. These two approaches, which coincide asymptotically, have similarities with line search and trust-region SQP methods.

Barrier methods for nonlinear optimization were developed in the 1960s but fell out of favor for almost two decades. The success of interior-point methods for linear programming stimulated renewed interest in them for the nonlinear case. By the late 1990s, a new generation of methods and software for nonlinear programming had emerged. Numerical experience indicates that interior-point methods are often faster than active-set SQP methods on large problems, particularly when the number of free variables is large. They may not yet be as robust, but significant advances are still being made in their design and implementation. The terms "interior-point methods" and "barrier methods" are now used interchangeably.

In Chapters 14 and 16 we discussed interior-point methods for linear and quadratic programming. It is not essential that the reader study those chapters before reading this one, although doing so will give a better perspective. The first part of this chapter assumes familiarity primarily with the KKT conditions and Newton's method, and the second part of the chapter relies on concepts from sequential quadratic programming presented in Chapter 18.

The problem under consideration in this chapter is written as follows:

$$\min_{x,s} \; f(x) \tag{19.1a}$$

$$\text{subject to} \quad c_{\mathrm{E}}(x) = 0, \tag{19.1b}$$

$$c_{\mathrm{I}}(x) - s = 0, \tag{19.1c}$$

$$s \geq 0. \tag{19.1d}$$

The vector $c_{\mathrm{I}}(x)$ is formed from the scalar functions $c_i(x),\ i \in \mathcal{I}$, and similarly for $c_{\mathrm{E}}(x)$. Note that we have transformed the inequalities $c_{\mathrm{I}}(x) \geq 0$ into equalities by the introduction of a vector s of slack variables. We use l to denote the number of equality constraints (that is, the dimension of the vector c_{E}) and m to denote the number of inequality constraints (the dimension of c_{I}).

19.1 TWO INTERPRETATIONS

Interior-point methods can be seen as continuation methods or as barrier methods. We discuss both derivations, starting with the continuation approach.

The KKT conditions (12.1) for the nonlinear program (19.1) can be written as

$$\nabla f(x) - A_{\mathrm{E}}{}^T(x)y - A_{\mathrm{I}}{}^T(x)z = 0, \tag{19.2a}$$

$$Sz - \mu e = 0, \tag{19.2b}$$

$$c_E(x) = 0, \tag{19.2c}$$
$$c_I(x) - s = 0, \tag{19.2d}$$

with $\mu = 0$, together with

$$s \geq 0, \qquad z \geq 0. \tag{19.3}$$

Here $A_E(x)$ and $A_I(x)$ are the Jacobian matrices of the functions c_E and c_I, respectively, and y and z are their Lagrange multipliers. We define S and Z to be the diagonal matrices whose diagonal entries are given by the vectors s and z, respectively, and let $e = (1, 1, \ldots, 1)^T$.

Equation (19.2b), with $\mu = 0$, and the bounds (19.3) introduce into the problem the combinatorial aspect of determining the optimal active set, illustrated in Example 15.1. We circumvent this difficulty by letting μ be strictly positive, thus forcing the variables s and z to take positive values. The homotopy (or continuation) approach consists of (approximately) solving the *perturbed KKT conditions* (19.2) for a sequence of positive parameters $\{\mu_k\}$ that converges to zero, while maintaining $s, z > 0$. The hope is that, in the limit, we will obtain a point that satisfies the KKT conditions for the nonlinear program (19.1). Furthermore, by requiring the iterates to decrease a merit function (or to be acceptable to a filter), the iteration is likely to converge to a minimizer, not simply a KKT point.

The homotopy approach is justified locally. In a neighborhood of a solution (x^*, s^*, y^*, z^*) that satisfies the linear independence constraint qualification (LICQ) (Definition 12.4), the strict complementarity condition (Definition 12.5), and the second-order sufficient conditions (Theorem 12.6), we have that for all sufficiently small positive values of μ, the system (19.2) has a locally unique solution, which we denote by $(x(\mu), s(\mu), y(\mu), z(\mu))$. The trajectory described by these points is called the *primal-dual central path*, and it converges to (x^*, s^*, y^*, z^*) as $\mu \to 0$.

The second derivation of interior-point methods associates with (19.1) the barrier problem

$$\min_{x,s} \; f(x) - \mu \sum_{i=1}^{m} \log s_i \tag{19.4a}$$
$$\text{subject to} \quad c_E(x) = 0, \tag{19.4b}$$
$$c_I(x) - s = 0, \tag{19.4c}$$

where μ is a positive parameter and $\log(\cdot)$ denotes the natural logarithm function. One need not include the inequality $s \geq 0$ in (19.4) because minimization of the barrier term $-\mu \sum_{i=1}^{m} \log s_i$ in (19.4a) prevents the components of s from becoming too close to zero. (Recall that $(-\log t) \to \infty$ as $t \downarrow 0$.) Problem (19.4) also avoids the combinatorial aspect of nonlinear programs, but its solution does not coincide with that of (19.1) for $\mu > 0$. The barrier approach consists of finding (approximate) solutions of the barrier problem (19.4) for a sequence of positive barrier parameters $\{\mu_k\}$ that converges to zero.

To compare the homotopy and barrier approaches, we write the KKT conditions for (19.4) as follows:

$$\nabla f(x) - A_{\text{E}}{}^T(x)y - A_{\text{I}}{}^T(x)z = 0, \tag{19.5a}$$

$$-\mu S^{-1}e + z = 0, \tag{19.5b}$$

$$c_{\text{E}}(x) = 0, \tag{19.5c}$$

$$c_{\text{I}}(x) - s = 0. \tag{19.5d}$$

Note that they differ from (19.2) only in the second equation, which becomes quite nonlinear near the solution as $s \to 0$. It is advantageous for Newton's method to transform the rational equation (19.5b) into a quadratic equation. We do so by multiplying this equation by S, a procedure that does not change the solution of (19.5) because the diagonal elements of S are positive. After this transformation, the KKT conditions for the barrier problem coincide with the perturbed KKT system (19.2).

The term "interior point" derives from the fact that early barrier methods [98] did not use slacks and assumed that the initial point x_0 is feasible with respect to the inequality constraints $c_i(x) \geq 0,\ i \in \mathcal{I}$. These methods used the barrier function

$$f(x) - \mu \sum_{i \in \mathcal{I}} \log c_i(x)$$

to prevent the iterates from leaving the feasible region defined by the inequalities. (We discuss this barrier function further in Section 19.6.) Most modern interior-point methods are infeasible (they can start from any initial point x_0) and remain interior only with respect to the constraints $s \geq 0, z \geq 0$. However, they can be designed so that once they generate a feasible iterate, all subsequent iterates remain feasible with respect to the inequalities.

In the next sections we will see that the homotopy and barrier interpretations are both useful. The homotopy view gives rise to the definition of the primal-dual direction, whereas the barrier view is vital in the design of globally convergent iterations.

19.2 A BASIC INTERIOR-POINT ALGORITHM

Applying Newton's method to the nonlinear system (19.2), in the variables x, s, y, z, we obtain

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & -A_{\text{E}}{}^T(x) & -A_{\text{I}}{}^T(x) \\ 0 & Z & 0 & S \\ A_{\text{E}}(x) & 0 & 0 & 0 \\ A_{\text{I}}(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ p_y \\ p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_{\text{E}}{}^T(x)y - A_{\text{I}}{}^T(x)z \\ Sz - \mu e \\ c_{\text{E}}(x) \\ c_{\text{I}}(x) - s \end{bmatrix}, \tag{19.6}$$

where $\mathcal{L}$ denotes the Lagrangian for (19.1a)–(19.1c):

$$\mathcal{L}(x, s, y, z) = f(x) - y^T c_{\mathrm{E}}(x) - z^T (c_{\mathrm{I}}(x) - s). \tag{19.7}$$

The system (19.6) is called the *primal-dual system* (in contrast with the primal system discussed in Section 19.3). After the step $p = (p_x, p_s, p_y, p_z)$ has been determined, we compute the new iterate (x^+, s^+, y^+, z^+) as

$$x^+ = x + \alpha_s^{\max} p_x, \quad s^+ = s + \alpha_s^{\max} p_s, \tag{19.8a}$$

$$y^+ = y + \alpha_z^{\max} p_y, \quad z^+ = z + \alpha_z^{\max} p_z, \tag{19.8b}$$

where

$$\alpha_s^{\max} = \max\{\alpha \in (0, 1] : s + \alpha p_s \geq (1 - \tau)s\}, \tag{19.9a}$$

$$\alpha_z^{\max} = \max\{\alpha \in (0, 1] : z + \alpha p_z \geq (1 - \tau)z\}, \tag{19.9b}$$

with $\tau \in (0, 1)$. (A typical value of τ is 0.995.) The condition (19.9), called the *fraction to the boundary rule*, prevents the variables s and z from approaching their lower bounds of 0 too quickly.

This simple iteration provides the basis of modern interior-point methods, though various modifications are needed to cope with nonconvexities and nonlinearities. The other major ingredient is the procedure for choosing the sequence of parameters $\{\mu_k\}$, which from now on we will call the *barrier parameters*. In the approach studied by Fiacco and McCormick [98], the barrier parameter μ is held fixed for a series of iterations until the KKT conditions (19.2) are satisfied to some accuracy. An alternative approach is to update the barrier parameter at each iteration. Both approaches have their merits and are discussed in Section 19.3.

The primal-dual matrix in (19.6) remains nonsingular as the iteration converges to a solution that satisfies the second-order sufficiency conditions and strict complementarity. More specifically, if x^* is a solution point for which strict complementarity holds, then for every index i either s_i or z_i remains bounded away from zero as the iterates approach x^*, ensuring that the second block row of the primal-dual matrix (19.6) has full row rank. Therefore, the interior-point approach does not, in itself, give rise to ill conditioning or singularity. This fact allows us to establish a fast (superlinear) rate of convergence; see Section 19.8.

We summarize the discussion by describing a concrete implementation of this basic interior-point method. We use the following error function, which is based on the perturbed KKT system (19.2):

$$\begin{aligned} E(x, s, y, z; \mu) = \max \big\{ &\|\nabla f(x) - A_{\mathrm{E}}(x)^T y - A_{\mathrm{I}}(x)^T z\|, \|Sz - \mu e\|, \\ &\|c_{\mathrm{E}}(x)\|, \|c_{\mathrm{I}}(x) - s\| \big\}, \end{aligned} \tag{19.10}$$

for some vector norm $\|\cdot\|$.

Algorithm 19.1 (Basic Interior-Point Algorithm).

Choose x_0 and $s_0 > 0$, and compute initial values for the multipliers y_0 and $z_0 > 0$. Select an initial barrier parameter $\mu_0 > 0$ and parameters $\sigma, \tau \in (0, 1)$. Set $k \leftarrow 0$.

repeat until a stopping test for the nonlinear program (19.1) is satisfied
  repeat until $E(x_k, s_k, y_k, z_k; \mu_k) \leq \mu_k$
    Solve (19.6) to obtain the search direction $p = (p_x, p_s, p_y, p_z)$;
    Compute $\alpha_s^{\max}, \alpha_z^{\max}$ using (19.9);
    Compute $(x_{k+1}, s_{k+1}, y_{k+1}, z_{k+1})$ using (19.8);
    Set $\mu_{k+1} \leftarrow \mu_k$ and $k \leftarrow k + 1$;
  end
  Choose $\mu_k \in (0, \sigma\mu_k)$;
end

An algorithm that updates the barrier parameter μ_k at every iteration is easily obtained from Algorithm 19.1 by removing the requirement that the KKT conditions be satisfied for each μ_k (the inner "repeat" loop) and by using a dynamic rule for updating μ_k in the penultimate line.

The following theorem provides a theoretical foundation for interior-point methods that compute only approximate solutions of the barrier problem.

Theorem 19.1.

Suppose that Algorithm 19.1 generates an infinite sequence of iterates $\{x_k\}$ and that $\{\mu_k\} \to 0$ (that is, that the algorithm does not loop infinitely in the inner "repeat" statement). Suppose that f and c are continuously differentiable functions. Then all limit points $\hat{x}$ of $\{x_k\}$ are feasible. Furthermore, if any limit point $\hat{x}$ of $\{x_k\}$ satisfies the linear independence constraint qualification (LICQ), then the first-order optimality conditions of the problem (19.1) hold at $\hat{x}$.

PROOF. For simplicity, we prove the result for the case in which the nonlinear program (19.1) contains only inequality constraints, leaving the extension of the result as an exercise. For ease of notation, we denote the inequality constraints c_i by c. Let $\hat{x}$ be a limit point of the sequence $\{x_k\}$, and let $\{x_{k_l}\}$ be a convergent subsequence, namely, $\{x_{k_l}\} \to \hat{x}$. Since $\mu_k \to 0$, the error E given by (19.10) converges to zero, so we have $(c_{k_l} - s_{k_l}) \to 0$. By continuity of c, this fact implies that $\hat{c} \stackrel{\text{def}}{=} c(\hat{x}) \geq 0$ (that is, $\hat{x}$ is feasible) and $s_{k_l} \to \hat{s} = \hat{c}$.

Now suppose that the linear independence constraint qualification holds at $\hat{x}$, and consider the set of active indices

$$\mathcal{A} = \{i : \hat{c}_i = 0\}.$$

For $i \notin \mathcal{A}$, we have $\hat{c}_i > 0$ and $\hat{s}_i > 0$, and thus by the complementarity condition (19.2b), we have that $[z_{k_l}]_i \to 0$. From this fact and $\nabla f_{k_l} - A_{k_l}^T z_{k_l} \to 0$, we deduce that

$$\nabla f_{k_l} - \sum_{i \in \mathcal{A}} [z_{k_l}]_i \nabla c_i(x_{k_l}) \to 0. \tag{19.11}$$

By the constraint qualification hypothesis, the vectors $\{\nabla \hat{c}_i : i \in \mathcal{A}\}$ are linearly independent. Hence, by (19.11) and continuity of $\nabla f(\cdot)$ and $\nabla c_{(i)}(\cdot)$, $i \in \mathcal{A}$, the positive sequence $\{z_{k_l}\}$ converges to some value $\hat{z} \geq 0$. Taking the limit in (19.11), we have that

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

We also have that $\hat{c}^T \hat{z} = 0$, completing the proof. □

Practical interior-point algorithms fall into two categories. The first builds on Algorithm 19.1, adding a line search and features to control the rate of decrease in the slacks s and multipliers z, and introducing modifications in the primal-dual sytem when negative curvature is encountered. The second category of algorithms, presented in Section 19.5, computes steps by minimizing a quadratic model of (19.4), subject to a trust-region constraint. The two approaches share many features described in the next section.

19.3 ALGORITHMIC DEVELOPMENT

We now discuss a series of modifications and extensions of Algorithm 19.1 that enable it to solve nonconvex nonlinear problems, starting from any initial estimate.

Often, the primal-dual system (19.6) is rewritten in the symmetric form

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & A_{\text{E}}{}^T(x) & A_{\text{I}}{}^T(x) \\ 0 & \Sigma & 0 & -I \\ A_{\text{E}}(x) & 0 & 0 & 0 \\ A_{\text{I}}(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ -p_y \\ -p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_{\text{E}}{}^T(x) y - A_{\text{I}}{}^T(x) z \\ z - \mu S^{-1} e \\ c_{\text{E}}(x) \\ c_{\text{I}}(x) - s \end{bmatrix}, \tag{19.12}$$

where

$$\Sigma = S^{-1} Z. \tag{19.13}$$

This formulation permits the use of a symmetric linear equations solver, which reduces the computational work of each iteration.

PRIMAL VS. PRIMAL-DUAL SYSTEM

If we apply Newton's method directly to the optimality conditions (19.5) of the barrier problem (instead of transforming to (19.5b) first) and then symmetrize the iteration matrix, we obtain the system (19.12) but with Σ given by

$$\Sigma = \mu S^{-2}. \tag{19.14}$$

This is often called the *primal* system, in contrast with the *primal-dual* system arising from (19.13). (This nomenclature owes more to the historical development of interior-point methods than to the concept of primal-dual iterations.) Whereas in the primal-dual choice (19.13) the vector z can be seen as a general multiplier estimate, the primal term (19.14) is obtained by making the specific selection $Z = \mu S^{-1}$; we return to this choice of multipliers in Section 19.6.

Even though the systems (19.2) and (19.5) are equivalent, Newton's method applied to them will generally produce different iterates, and there are reasons for preferring the primal-dual system. Note that (19.2b) has the advantage that its derivatives are bounded as any slack variables approach zero; such is not the case with (19.5b). Moreover, analysis of the primal step as well as computational experience has shown that, under some circumstances, the primal step (19.12), (19.14) tends to produce poor steps that violate the bounds $s > 0$ and $z > 0$ significantly, resulting in slow progress; see Section 19.6.

SOLVING THE PRIMAL-DUAL SYSTEM

Apart from the cost of evaluating the problem functions and their derivatives, the work of the interior-point iteration is dominated by the solution of the primal-dual system (19.12), (19.13). An efficient linear solver, using either sparse factorization or iterative techniques, is therefore essential for fast solution of large problems.

The symmetric matrix in (19.12) has the familiar form of a KKT matrix (cf. (16.7), (18.6)), and the linear system can be solved by the approaches described in Chapter 16. We can first reduce the system by eliminating p_s using the second equation in (19.6), giving

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & A_E{}^T(x) & A_I{}^T(x) \\ A_E(x) & 0 & 0 \\ A_I(x) & 0 & -\Sigma^{-1} \end{bmatrix} \begin{bmatrix} p_x \\ -p_y \\ -p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_E{}^T(x)y - A_I{}^T(x)z \\ c_E(x) \\ c_I(x) - \mu Z^{-1}e \end{bmatrix}. \tag{19.15}$$

This system can be factored by using a symmetric indefinite factorization; see (16.12). If we denote the coefficient matrix in (19.15) by K, this factorization computes $P^T K P = LBL^T$, where L is lower triangular and B is block diagonal, with blocks of size 1×1 or 2×2. P is a matrix of row and column permutations that seeks a compromise between the goals of preserving sparsity and ensuring numerical stability; see (3.51) and the discussion that follows.

The system (19.15) can be reduced further by eliminating p_z using the last equation, to obtain the condensed coefficient matrix

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} + A_{\mathrm{I}}{}^T \Sigma A_{\mathrm{I}} & A_{\mathrm{E}}{}^T(x) \\ A_{\mathrm{E}}(x) & 0 \end{bmatrix}, \tag{19.16}$$

which is much smaller than (19.12) when the number of inequality constraints is large. Although significant fill-in can arise from the term $A_{\mathrm{I}}{}^T \Sigma A_{\mathrm{I}}$, it is tolerable in many applications. A particularly favorable case, in which $A_{\mathrm{I}}{}^T \Sigma A_{\mathrm{I}}$ is diagonal, arises when the inequality constraints are simple bounds.

The primal-dual system in any of the symmetric forms (19.12), (19.15), (19.16) is ill conditioned because, by (19.13), some of the elements of Σ diverge to ∞, while others converge to zero as $\mu \to 0$. Nevertheless, because of the special form in which this ill conditioning arises, the direction computed by a stable direct factorization method is usually accurate. Damaging errors result only when the slacks s or multipliers z become very close to zero (or when the Hessian $\nabla^2_{xx}\mathcal{L}$ or the Jacobian matrix A_{E} is almost rank deficient). For this reason, direct factorization techniques are considered the most reliable techniques for computing steps in interior-point methods.

Iterative linear algebra techniques can also be used for the step computation. Ill conditioning is a grave concern in this context, and preconditioners that cluster the eigenvalues of Σ must be used. Fortunately, such preconditioners are easy to construct. For example, let us introduce the change of variables $\tilde{p}_s = S^{-1}p_s$ in the system (19.12), and multiply the second equation in (19.12) by S, transforming the term Σ into $S\Sigma S$. As $\mu \to 0$ (and assuming that $SZ \approx \mu I$) we have from (19.13) that all the elements of $S\Sigma S$ cluster around μI. Other scalings can be used as well. The change of variables $\tilde{p}_s = \Sigma^{1/2}p_s$ provides the perfect preconditioner, while $\tilde{p}_s = \sqrt{\mu}S^{-1}p_s$ transforms Σ to $S\Sigma S/\mu$, which converges to I as $\mu \to 0$.

We can apply an iterative method to one of the symmetric indefinite systems (19.12), (19.15), or (19.16). The conjugate gradient method is not appropriate (except as explained below) because it is designed for positive definite systems, but we can use GMRES, QMR, or LSQR (see [136]). In addition to employing preconditioning that removes the ill conditioning caused by the barrier approach, as discussed above, we need to deal with possible ill conditioning caused by the Hessian $\nabla^2_{xx}\mathcal{L}$ or the Jacobian matrices A_{E} and A_{I}. General-purpose preconditioners are difficult to find in this context, and the success of an iterative method hinges on the use of problem-specific or structured preconditioners.

An effective alternative is to use a null-space approach to solve the primal-dual system and apply the CG method in the (positive definite) reduced space. As explained in Section 16.3, we can do this by applying the *projected CG iteration* of Algorithm 16.2 using a so-called constraint preconditioner. In the context of the system (19.12) the preconditioner

has the form

$$\begin{bmatrix} G & 0 & A_E^T(x) & A_I^T(x) \\ 0 & T & 0 & -I \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -I & 0 & 0 \end{bmatrix}, \tag{19.17}$$

where G is a sparse matrix that is positive definite on the null space of the constraints and T is a diagonal matrix that equals or approximates Σ. This preconditioner keeps the Jacobian information of A_E and A_I intact and thereby removes any ill conditioning present in these matrices.

UPDATING THE BARRIER PARAMETER

The sequence of barrier parameters $\{\mu_k\}$ must converge to zero so that, in the limit, we recover the solution of the nonlinear programming problem (19.1). If μ_k is decreased too slowly, a large number of iterations will be required for convergence; but if it is decreased too quickly, some of the slacks s or multipliers z may approach zero prematurely, slowing progress of the iteration. We now describe several techniques for updating μ_k that have proved to be effective in practice.

The strategy implemented in Algorithm 19.1, which we call the *Fiacco–McCormick approach*, fixes the barrier parameter until the perturbed KKT conditions (19.2) are satisfied to some accuracy. Then the barrier parameter is decreased by the rule

$$\mu_{k+1} = \sigma_k \mu_k, \quad \text{with} \quad \sigma_k \in (0, 1). \tag{19.18}$$

Some early implementations of interior-point methods chose σ_k to be a constant (for example, $\sigma_k = 0.2$). It is, however, preferable to let σ_k take on two or more values (for example, 0.2 and 0.1), choosing smaller values when the most recent iterations make significant progress toward the solution. Furthermore, by letting $\sigma_k \to 0$ near the solution, and letting the parameter τ in (19.9) converge to 1, a superlinear rate of convergence can be obtained.

The Fiacco–McCormick approach works well on many problems, but it can be sensitive to the choice of the initial point, the initial barrier parameter value, and the scaling of the problem.

Adaptive strategies for updating the barrier parameter are more robust in difficult situations. These strategies, unlike the Fiacco–McCormick approach, vary μ at every iteration depending on the progress of the algorithm. Most such strategies are based on complementarity, as in the linear programming case (see Framework 14.1), and have the form

$$\mu_{k+1} = \sigma_k \frac{s_k^T z_k}{m}, \tag{19.19}$$

which allows μ_k to reflect the scale of the problem. One choice of σ_k, implemented in the LOQO package [294], is based on the deviation of the smallest complementarity product $[s_k]_i[z_k]_i$ from the average:

$$\sigma_k = 0.1 \min\left(0.05\frac{1-\xi_k}{\xi_k}, 2\right)^3, \quad \text{where} \quad \xi_k = \frac{\min_i [s_k]_i [z_k]_i}{(s^k)^T z^k/m}. \tag{19.20}$$

Here $[s_k]_i$ denotes the ith component of the iterate s_k, and similarly for $[z_k]_i$. When $\xi_k \approx 1$ (all the individual products are near to their average), the barrier parameter is decreased aggressively.

Predictor or probing strategies (see Section 14.2) can also be used to determine the parameter σ_k in (19.19). We calculate a predictor (affine scaling) direction

$$(\Delta x^{\text{aff}}, \Delta s^{\text{aff}}, \Delta y^{\text{aff}}, \Delta z^{\text{aff}})$$

by setting $\mu = 0$ in (19.12). We probe this direction by finding α_p^{aff} and α_d^{aff} to be the longest step lengths that can be taken along the affine scaling direction before violating the nonnegativity conditions $(s, z) \geq 0$. Explicit formulas for these step lengths are given by (19.9) with $\tau = 1$. We then define μ_{aff} to be the value of complementarity along the (shortened) affine scaling step, that is,

$$\mu_{\text{aff}} = (s_k + \alpha_s^{\text{aff}} \Delta s^{\text{aff}})^T (z_k + \alpha_z^{\text{aff}} \Delta z^{\text{aff}})/m, \tag{19.21}$$

and define σ_k as follows:

$$\sigma_k = \left(\frac{\mu_{\text{aff}}}{s_k^T z_k/m}\right)^3. \tag{19.22}$$

This heuristic choice of σ_k was proposed for linear programming problems (see (14.34)) and also works well for nonlinear programs.

HANDLING NONCONVEXITY AND SINGULARITY

The direction defined by the primal-dual system (19.12) is not always productive because it seeks to locate only KKT points; it can move toward a maximizer or other stationary points. In Chapter 18 we have seen that the Newton step (18.9) for the equality-constrained problem (18.1) can be guaranteed to be a descent direction for a large class of merit functions—and to be a productive direction for a filter—if the Hessian W is positive definite on the tangent space of the constraints. The reason is that, in this case, the step can be interpreted as the minimization of a convex model in the reduced space obtained by eliminating the linearized constraints.

For the primal-dual system (19.12), the step p is a descent direction if the matrix

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 \\ 0 & \Sigma \end{bmatrix} \tag{19.23}$$

is positive definite on the null space of the constraint matrix

$$\begin{bmatrix} A_{\mathrm{E}}(x) & 0 \\ A_{\mathrm{I}}(x) & -I \end{bmatrix}.$$

Lemma 16.3 states that this positive definiteness condition holds if the inertia of the primal-dual matrix in (19.12) is given by

$$(n+m, l+m, 0), \tag{19.24}$$

in other words, if this matrix has exactly $n+m$ positive, $l+m$ negative, and no zero eigenvalues. (Recall that l and m denote the number of equality and inequality constraints, respectively.) As discussed in Section 3.4, the inertia can be obtained from the symmetric-indefinite factorization of (19.12).

If the primal-dual matrix does not have the desired inertia, we can modify it as follows. Note that the diagonal matrix Σ is positive definite by construction but $\nabla^2_{xx}\mathcal{L}$ can be indefinite. Therefore, we can replace the latter matrix by $\nabla^2_{xx}\mathcal{L} + \delta I$, where $\delta > 0$ is sufficiently large to ensure that the inertia is given by (19.24). The size of this modification is not known beforehand, but we can try successively larger values of δ until the desired inertia is obtained.

We must also guard against singularity of the primal-dual matrix caused by the rank deficiency of A_{E} (the matrix $[A_{\mathrm{I}} \;\; -I]$ always has full rank). We do so by including a regularization parameter $\gamma \geq 0$, in addition to the modification term δI, and work with the modified primal-dual matrix

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} + \delta I & 0 & A_{\mathrm{E}}(x)^T & A_{\mathrm{I}}(x)^T \\ 0 & \Sigma & 0 & -I \\ A_{\mathrm{E}}(x) & 0 & -\gamma I & 0 \\ A_{\mathrm{I}}(x) & -I & 0 & 0 \end{bmatrix}. \tag{19.25}$$

A procedure for selecting γ and δ is given in Algorithm B.1 in Appendix B. It is invoked at every iteration of the interior-point method to enforce the inertia condition (19.24) and to guarantee nonsingularity. Other matrix modifications to ensure positive definiteness have been discussed in Chapter 3 in the context of unconstrained minimization.

STEP ACCEPTANCE: MERIT FUNCTIONS AND FILTERS

The role of the merit function or filter is to determine whether a step is productive and should be accepted. Since interior-point methods can be seen as methods for solving the barrier problem (19.4), it is appropriate to define the merit function ϕ or filter in terms of barrier functions. We may use, for example, an exact merit function of the form

$$\phi_\nu(x, s) = f(x) - \mu \sum_{i=1}^{m} \log s_i + \nu \|c_{\mathrm{E}}(x)\| + \nu \|c_{\mathrm{I}}(x) - s\|, \tag{19.26}$$

where the norm is chosen, say, to be the ℓ_1 or the ℓ_2 norm (unsquared). The penalty parameter $\nu > 0$ can be updated by using the strategies described in Chapter 18.

In a line search method, after the step p has been computed and the maximum step lengths (19.9) have been determined, we perform a backtracking line search that computes the step lengths

$$\alpha_s \in (0, \alpha_s^{\max}], \qquad \alpha_z \in (0, \alpha_z^{\max}], \tag{19.27}$$

providing sufficient decrease of the merit function or ensuring acceptability by the filter. The new iterate is then defined as

$$x^+ = x + \alpha_s p_x, \quad s^+ = s + \alpha_s p_s, \tag{19.28a}$$

$$y^+ = y + \alpha_z p_y, \quad z^+ = z + \alpha_z p_z. \tag{19.28b}$$

When defining a filter (see Section 15.4) the pairs of the filter are formed, on the one hand, by the values of the barrier function $f(x) - \mu \sum_{i=1}^{m} \log s_i$ and, on the other hand, by the constraint violations $\|(c_{\mathrm{E}}(x), c_{\mathrm{I}}(x) - s)\|$. A step will be accepted if it is not dominated by any element in the filter. Under certain circumstances, if the step is not accepted by the filter, instead of reducing the step length α_s in (19.8a), a feasibility restoration phase is invoked; see the Notes and References at the end of the chapter.

QUASI-NEWTON APPROXIMATIONS

A quasi-Newton version of the primal-dual step is obtained by replacing $\nabla^2_{xx}\mathcal{L}$ in (19.12) by a quasi-Newton approximation B. We can use the BFGS (6.19) or SR1 (6.24) update formulas described in Chapter 6 to define B, or we can follow a limited-memory BFGS approach (see Chapter 7). It is important to approximate the Hessian of the Lagrangian of the nonlinear program, not the Hessian of the barrier function, which is highly ill conditioned and changes rapidly.

The correction pairs used by the quasi-Newton updating formula are denoted here by $(\Delta x, \Delta l)$, replacing the notation (s, y) of Chapter 6. After computing a step from (x, s, y, z)

to (x^+, s^+, y^+, z^+), we define

$$\Delta l = \nabla_x \mathcal{L}(x^+, s^+, y^+, z^+) - \nabla_x \mathcal{L}(x, s^+, y^+, z^+),$$
$$\Delta x = x^+ - x.$$

To ensure that the BFGS method generates a positive definite matrix, one can skip or damp the update; see (18.14) and (18.15). SR1 updating must be safeguarded to avoid unboundedness, as discussed in Section 6.2, and may also need to be modified so that the inertia of the primal-dual matrix is given by (19.24). This modification can be performed by means of Algorithm B.1.

The quasi-Newton matrices B generated in this manner are dense $n \times n$ matrices. For large problems, limited-memory updating is desirable. One option is to implement a limited-memory BFGS method by using the compact representations described in Section 7.2. Here B has the form

$$B = \xi I + W M W^T, \tag{19.29}$$

where $\xi > 0$ is a scaling factor, W is an $n \times 2\hat{m}$ matrix, M is a $2\hat{m} \times 2\hat{m}$ symmetric and nonsingular matrix, and $\hat{m}$ denotes the number of correction pairs saved in the limited-memory updating procedure. The matrices W and M are formed by using the vectors $\{\Delta l^k\}$ and $\{\Delta x^k\}$ accumulated in the last $\hat{m}$ iterations. Since the limited-memory matrix B is positive definite, and assuming A_E has full rank, the primal-dual matrix is nonsingular, and we can compute the solution to (19.12) by inverting the coefficient matrix using the Sherman–Morrison–Woodbury formula (see Exercise 19.14).

FEASIBLE INTERIOR-POINT METHODS

In many applications, it is desirable for all of the iterates generated by an optimization algorithm to be feasible with respect to some or all of the *inequality* constraints. For example, the objective function may be defined only when some of the constraints are satisfied, making this feature essential.

Interior-point methods provide a natural framework for deriving feasible algorithms. If the current iterate x satisfies $c_I(x) > 0$, then it is easy to adapt the primal-dual iteration (19.12) so that feasibility is preserved. After computing the step p, we let $x^+ = x + p_x$, redefine the slacks as

$$s^+ \leftarrow c_I(x^+), \tag{19.30}$$

and test whether the point (x^+, s^+) is acceptable for the merit function ϕ. If so, we define this point to be the new iterate; otherwise we reject the step p and compute a new, shorter trial step. In a line search algorithm we backtrack, and in a trust-region method we compute a new step with a reduced trust-region bound. This strategy is justified by the fact that if at a trial point we have that $c_i(x^+) \leq 0$ for some inequality constraint, the value of the merit

function is $+\infty$, and we reject the trial point. We will also reject steps $x + p_x$ that are too close to the boundary of the feasible region because such steps increase the barrier term $-\mu \sum_{i \in \mathcal{I}} \log(s_i)$ in the merit function (19.26).

Making the substitution (19.30) has the effect of replacing $\log(s_i)$ with $\log(c_i(x))$ in the merit function, a technique reminiscent of the classical primal log-barrier approach discussed in Section 19.6.

19.4 A LINE SEARCH INTERIOR-POINT METHOD

We now give a more detailed description of a line search interior-point method. We denote by $D\phi(x, s; p)$ the directional derivative of the merit function ϕ_v at (x, s) in the direction p. The stopping conditions are based on the error function (19.10).

Algorithm 19.2 (Line Search Interior-Point Algorithm).

Choose x_0 and $s_0 > 0$, and compute initial values for the multipliers y_0 and $z_0 > 0$. If a quasi-Newton approach is used, choose an $n \times n$ symmetric and positive definite initial matrix B_0. Select an initial barrier parameter $\mu > 0$, parameters $\eta, \sigma \in (0, 1)$, and tolerances ϵ_μ and ϵ_{TOL}. Set $k \leftarrow 0$.

repeat until $E(x_k, s_k, y_k, z_k; 0) \leq \epsilon_{\text{TOL}}$
  repeat until $E(x_k, s_k, y_k, z_k; \mu) \leq \epsilon_\mu$
    Compute the primal-dual direction $p = (p_x, p_s, p_y, p_z)$ from
      (19.12), where the coefficient matrix is modified as in
      (19.25), if necessary;
    Compute $\alpha_s^{\max}, \alpha_z^{\max}$ using (19.9); Set $p_w = (p_x, p_s)$;
    Compute step lengths α_s, α_z satisfying both (19.27) and
      $\phi_v(x_k + \alpha_s p_x, s_k + \alpha_s p_s) \leq \phi_v(x_k, s_k) + \eta \alpha_s D\phi_v(x_k, s_k; p_w)$;
    Compute $(x_{k+1}, s_{k+1}, y_{k+1}, z_{k+1})$ using (19.28);
    if a quasi-Newton approach is used
      update the approximation B_k;
    Set $k \leftarrow k + 1$;
  end
  Set $\mu \leftarrow \sigma \mu$ and update ϵ_μ;
end

The barrier tolerance can be defined, for example, as $\epsilon_\mu = \mu$, as in Algorithm 19.1. An adaptive strategy that updates the barrier parameter μ at every step is easily implemented in this framework. If the merit function can cause the Maratos effect (see Section 15.4), a second-order correction or a nonmonotone strategy should be implemented. An alternative to using a merit function is to employ a filter mechanism to perform the line search.

We will see in Section 19.7 that Algorithm 19.2 must be safeguarded to ensure global convergence.

19.5 A TRUST-REGION INTERIOR-POINT METHOD

We now consider an interior-point method that uses trust regions to promote convergence. As in the unconstrained case, the trust-region formulation allows great freedom in the choice of the Hessian and provides a mechanism for coping with Jacobian and Hessian singularities. The price to pay for this flexibility is a more complex iteration than in the line search approach.

The interior-point method described below is asymptotically equivalent to the line search method discussed in Section 19.4, but differs significantly in two respects. First, it is not fully a primal-dual method in the sense that it first computes a step in the variables (x, s) and then updates the estimates for the multipliers, as opposed to the approach of Algorithm 19.1, in which primal and dual variables are computed simultaneously. Second, the trust-region method uses a scaling of the variables that discourages moves toward the boundary of the feasible region. This causes the algorithm to generate steps that can be different from, and enjoy more favorable convergence properties than, those produced by a line search method.

We first describe a trust-region algorithm for finding approximate solutions of a fixed barrier problem. We then present a complete interior-point method in which the barrier parameter is driven to zero.

AN ALGORITHM FOR SOLVING THE BARRIER PROBLEM

The barrier problem (19.4) is an equality-constrained optimization problem and can be solved by using a sequential quadratic programming method with trust regions. A straightforward application of SQP techniques to the barrier problem leads, however, to inefficient steps that tend to violate the positivity of the slack variables and are frequently cut short by the trust-region constraint. To overcome this problem, we design an SQP method tailored to the structure of barrier problems.

At the iterate (x, s), and for a given barrier parameter μ, we first compute Lagrange multiplier estimates (y, z) and then compute a step $p = (p_x, p_s)$ that approximately solves the subproblem

$$\begin{aligned}
\min_{p_x, p_s} \quad & \nabla f^T p_x + \frac{1}{2} p_x^T \nabla^2_{xx} \mathcal{L} p_x - \mu e^T S^{-1} p_s + \frac{1}{2} p_s^T \Sigma p_s && \text{(19.31a)} \\
\text{subject to} \quad & A_{\text{E}}(x) p_x + c_{\text{E}}(x) = r_{\text{E}}, && \text{(19.31b)} \\
& A_{\text{I}}(x) p_x - p_s + (c_{\text{I}}(x) - s) = r_{\text{I}}, && \text{(19.31c)} \\
& \|(p_x, S^{-1} p_s)\|_2 \leq \Delta, && \text{(19.31d)} \\
& p_s \geq -\tau s. && \text{(19.31e)}
\end{aligned}$$

Here Σ is the primal-dual matrix (19.13), and the scalar $\tau \in (0, 1)$ is chosen close to 1 (for example, 0.995). The inequality (19.31e) plays the same role as the fraction to the boundary rule (19.9). Ideally, we would like to set $r = (r_E, r_I) = 0$, but since this can cause the constraints (19.31b)–(19.31d) to be incompatible or to give a step p that makes little progress toward feasibility, we choose the parameter r by an auxiliary computation, as in Algorithm 18.4.

We motivate the choice of the objective (19.31a) by noting that the first-order optimality conditions of (19.31a)–(19.31c) are given by (19.2) (with the second block of equations scaled by S^{-1}). Thus the step computed from the subproblem (19.31) is related to the primal-dual line search step in the same way as the SQP and Newton–Lagrange steps of Section 18.1.

The trust-region constraint (19.31d) guarantees that the problem (19.31) has a finite solution even when $\nabla^2_{xx}\mathcal{L}(x, s, y, z)$ is not positive definite, and therefore this Hessian need never be modified. In addition, the trust-region formulation ensures that adequate progress is made at every iteration. To justify the scaling S^{-1} used in (19.31d), we note that the shape of the trust region must take into account the requirement that the slacks not approach zero prematurely. The scaling S^{-1} serves this purpose because it restricts those components i of the step vector p_s for which s_i is close to its lower bound of zero. As we see below, it also plays an important role in the choice of the relaxation vectors r_E and r_I.

We outline this SQP trust-region approach as follows. The stopping condition is defined in terms of the error function E given by (19.10), and the merit function ϕ_ν can be defined as in (19.26) using the 2-norm, $\|\cdot\|_2$.

Algorithm 19.3 (Trust-Region Algorithm for Barrier Problems).

Input parameters: $\mu > 0$, x_0, $s_0 > 0$, ϵ_μ, and $\Delta_0 > 0$. Compute Lagrange multiplier estimates y_0 and $z_0 > 0$. Set $k \leftarrow 0$.

repeat until $E(x_k, s_k, y_k, z_k; \mu) \le \epsilon_\mu$
  Compute $p = (p_x, p_s)$ by approximately solving (19.31).
  if p provides sufficient decrease in the merit function ϕ_ν
    Set $x_{k+1} \leftarrow x_k + p_x$, $s_{k+1} \leftarrow s_k + p_s$;
    Compute new multiplier estimates $y_{k+1}, z_{k+1} > 0$
      and set $\Delta_{k+1} \ge \Delta_k$;
  else
    Define $x_{k+1} \leftarrow x_k$, $s_{k+1} \leftarrow s_k$, and set $\Delta_{k+1} < \Delta_k$;
  end
  Set $k \leftarrow k + 1$;
end (repeat)

Algorithm 19.3 is applied for a fixed value of the barrier parameter μ. A complete interior-point algorithm driven by a sequence $\{\mu_k\} \to 0$ is described below. First, we discuss how to find an approximate solution of the subproblem (19.31), along with Lagrange multiplier estimates (y_{k+1}, z_{k+1}).

STEP COMPUTATION

The subproblem (19.31a)–(19.31e) is difficult to minimize exactly because of the presence of the nonlinear constraint (19.31d) and the bounds (19.31e). An important observation is that we can compute useful *inexact* solutions, at moderate cost. Since this approach scales up well with the number of variables and constraints, it provides a framework for developing practical interior-point methods for large-scale optimization.

The first step in the solution process is to make a change of variables that transforms the trust-region constraint (19.31d) into a ball. By defining

$$\tilde{p} = \begin{bmatrix} p_x \\ \tilde{p}_s \end{bmatrix} = \begin{bmatrix} p_x \\ S^{-1} p_s \end{bmatrix}, \tag{19.32}$$

we can write problem (19.31) as

$$\begin{aligned}
\min_{p_x, \tilde{p}_s} \quad & \nabla f^T p_x + \frac{1}{2} p_x^T \nabla^2_{xx} \mathcal{L} p_x - \mu e^T \tilde{p}_s + \frac{1}{2} \tilde{p}_s^T S \Sigma S \tilde{p}_s && \text{(19.33a)} \\
\text{subject to} \quad & A_{\text{E}}(x) p_x + c_{\text{E}}(x) = r_{\text{E}}, && \text{(19.33b)} \\
& A_{\text{I}}(x) p_x - S \tilde{p}_s + (c_{\text{I}}(x) - s) = r_{\text{I}}, && \text{(19.33c)} \\
& \|(p_x, \tilde{p}_s)\|_2 \leq \Delta, && \text{(19.33d)} \\
& \tilde{p}_s \geq -\tau e. && \text{(19.33e)}
\end{aligned}$$

To compute the vectors r_{E} and r_{I}, we proceed as in Section 18.5 and formulate the following *normal subproblem* in the variable $v = (v_x, v_s)$:

$$\begin{aligned}
\min_{v} \quad & \|A_{\text{E}}(x) v_x + c_{\text{E}}(x)\|_2^2 + \|A_{\text{I}}(x) v_x - S v_s + (c_{\text{I}}(x) - s)\|_2^2 && \text{(19.34a)} \\
\text{subject to} \quad & \|(v_x, v_s)\|_2 \leq 0.8\Delta, && \text{(19.34b)} \\
& v_s \geq -(\tau/2) e. && \text{(19.34c)}
\end{aligned}$$

If we ignore (19.34c), this problem has the standard form of a trust-region problem, and we can compute an approximate solution by using the techniques discussed in Chapter 4, such as the dogleg method. If the solution violates the bounds (19.34c), we can backtrack so that these bounds are satisfied.

Having solved (19.34), we define the vectors r_{E} and r_{I} in (19.33b)–(19.33c) to be the residuals in the normal step computation, namely,

$$r_{\text{E}} = A_{\text{E}}(x) v_x + c_{\text{E}}(x), \qquad r_{\text{I}} = A_{\text{I}}(x) v_x - S v_s + (c_{\text{I}}(x) - s). \tag{19.35}$$

We are now ready to compute an approximate solution $\tilde{d}$ of the subproblem (19.33). By (19.35), the vector v is a particular solution of the linear constraints (19.33b)–(19.33c). We

can then solve the equality-constrained quadratic program (19.33a)–(19.33c) by using the projected conjugate gradient iteration given in Algorithm 16.2. We terminate the projected CG iteration by Steihaug's rules: During the solution by CG we monitor the satisfaction of the trust-region constraint (19.33d) and stop if the boundary of this region is reached, if negative curvature is detected, or if an approximate solution is obtained. If the solution given by the projected CG iteration does not satisfy the bounds (19.33e), we backtrack so that they are satisfied. After the step $(p_x, \tilde{p}_s)$ has been computed, we recover p from (19.32).

As discussed in Section 16.3, every iteration of the projected CG iteration requires the solution of a linear system in order to perform the projection operation. For the quadratic program (19.33a)–(19.33c) this projection matrix is given by

$$\begin{bmatrix} I & \hat{A}^T \\ \hat{A} & 0 \end{bmatrix}, \quad \text{with} \quad \hat{A} = \begin{bmatrix} A_{\mathrm{E}}(x) & 0 \\ A_{\mathrm{I}}(x) & -S \end{bmatrix}. \tag{19.36}$$

Thus, although this trust-region approach still requires the solution of an augmented system, the matrix (19.36) is simpler than the primal-dual matrix (19.12). In particular, the Hessian $\nabla^2_{xx}\mathcal{L}$ need never be factored because the CG approach requires only products of this matrix with vectors.

We mentioned in Section 19.3 that the term $S\Sigma S$ in (19.33a) has a much tighter distribution of eigenvalues than Σ. Therefore the CG method will normally not be adversely affected by ill conditioning and is a viable approach for solving the quadratic program (19.33a)–(19.33c).

LAGRANGE MULTIPLIERS ESTIMATES AND STEP ACCEPTANCE

At an iterate (x, s), we choose (y, z) to be the least-squares multipliers (see (18.21)) corresponding to (19.33a)–(19.33c). We obtain the formula

$$\begin{bmatrix} y \\ z \end{bmatrix} = \left(\hat{A}\hat{A}^T\right)^{-1} \hat{A} \begin{bmatrix} \nabla f(x) \\ -\mu e \end{bmatrix}, \tag{19.37}$$

where $\hat{A}$ is given by (19.36) The multiplier estimates z obtained in this manner may not always be positive; to enforce positivity, we may redefine them as

$$z_i \leftarrow \min(10^{-3}, \mu/s_i), \qquad i = 1, 2, \ldots, m. \tag{19.38}$$

The quantity μ/s_i is called the ith primal multiplier estimate because if all components of z were defined by (19.38), then Σ would reduce to the primal choice, (19.14).

As is standard in trust-region methods, the step p is accepted if

$$\text{ared}(p) \geq \eta \, \text{pred}(p), \tag{19.39}$$

where

$$\text{ared}(p) = \phi_\nu(x, s) - \phi_\nu(x + p_x, s + p_s) \tag{19.40}$$

and where η is a constant in $(0, 1)$ (say, $\eta = 10^{-8}$). The predicted reduction is defined as

$$\text{pred}(p) = q_\nu(0) - q_\nu(p), \tag{19.41}$$

where q_ν is defined as

$$q_\nu(p) = \nabla f^T p_x + \frac{1}{2} p_x^T \nabla^2_{xx} \mathcal{L} p_x - \mu e^T S^{-1} p_s + \frac{1}{2} p_s^T \Sigma p_s + \nu m(p),$$

and

$$m(p) = \left\| \begin{bmatrix} A_{\text{E}}(x) p_x + c_{\text{E}}(x) \\ A_{\text{I}}(x) p_x - p_s + c_{\text{I}}(x) - s \end{bmatrix} \right\|_2.$$

To determine an appropriate value of the penalty parameter ν, we require that ν be large enough that

$$\text{pred}(p) \geq \rho \nu (m(0) - m(p)), \tag{19.42}$$

for some parameter $\rho \in (0, 1)$. This is the same as condition (18.35) used in Section 18.5, and the value of ν can be computed by the procedure described in that section.

DESCRIPTION OF A TRUST-REGION INTERIOR-POINT METHOD

We now present a more detailed description of the trust-region interior-point algorithm for solving the nonlinear programming problem (19.1). For concreteness we follow the Fiacco–McCormick strategy for updating the barrier parameter. The stopping conditions are stated, once more, in terms of the error function E defined by (19.10). In a quasi-Newton approach, the Hessian $\nabla^2_{xx} \mathcal{L}$ is replaced by a symmetric approximation.

Algorithm 19.4 (Trust-Region Interior-Point Algorithm).

Choose a value for the parameters $\eta > 0$, $\tau \in (0, 1)$, $\sigma \in (0, 1)$, and $\zeta \in (0, 1)$, and select the stopping tolerances ϵ_μ and ϵ_{TOL}. If a quasi-Newton approach is used, select an $n \times n$ symmetric initial matrix B_0. Choose initial values for $\mu > 0$, x_0, $s_0 > 0$, and Δ_0. Set $k \leftarrow 0$.

repeat until $E(x_k, s_k, y_k, z_k; 0) \leq \epsilon_{\text{TOL}}$
  repeat until $E(x_k, s_k, y_k, z_k; \mu) \leq \epsilon_\mu$
    Compute Lagrange multipliers from (19.37)–(19.38);

```
        Compute ∇²_xx L(x_k, s_k, y_k, z_k) or upate a quasi-Newton
              approximation B_k, and define Σ_k by (19.13);
        Compute the normal step v_k = (v_x, v_s);
        Compute p̃_k by applying the projected CG method to (19.33);
        Obtain the total step p_k from (19.32);
        Update ν_k to satisfy (19.42);
        Compute pred_k(p_k) by (19.41) and ared_k(p_k) by (19.40);
        if ared_k(p_k) ≥ η pred_k(p_k)
              Set x_{k+1} ← x_k + p_x, s_{k+1} ← s_k + p_s;
              Choose Δ_{k+1} ≥ Δ_k;
        else
              set x_{k+1} = x_k, s_{k+1} = s_k; and choose Δ_{k+1} < Δ_k;
        endif
        Set k ← k + 1;
    end
    Set μ ← σμ and update ε_μ;
end
```

The merit function (19.26) can reject steps that make good progress toward a solution: the Maratos effect discussed in Chapter 18. This deficiency can be overcome by selective application of a second-order correction step; see Section 15.4.

Algorithm 19.4 can easily be modified to implement an adaptive barrier update strategy. The barrier stop tolerance can be defined as $\epsilon_\mu = \mu$. Algorithm 19.4 is the basis of the KNITRO/CG method [50], which implements both exact Hessian and quasi-Newton options.

19.6 THE PRIMAL LOG-BARRIER METHOD

Prior to the introduction of primal-dual interior methods, barrier methods worked in the space of primal variables x. As in the quadratic penalty function approach of Chapter 17, the goal was to solve nonlinear programming problems by unconstrained minimization applied to a parametric sequence of functions.

Primal barrier methods are more easily described in the context of inequality-constrained problems of the form

$$\min_x \ f(x) \quad \text{subject to } c(x) \geq 0. \tag{19.43}$$

The log-barrier function is defined by

$$P(x; \mu) = f(x) - \mu \sum_{i \in \mathcal{I}} \log c_i(x), \tag{19.44}$$

where $\mu > 0$. One can show that the minimizers of $P(x; \mu)$, which we denote by $x(\mu)$, approach a solution of (19.43) as $\mu \downarrow 0$, under certain conditions; see, for example, [111]. The trajectory $\mathcal{C}_{\rm p}$ defined by

$$\mathcal{C}_{\rm p} \stackrel{\rm def}{=} \{x(\mu) \mid \mu > 0\} \tag{19.45}$$

is often referred to as the *primal central path.*

Since the minimizer $x(\mu)$ of $P(x; \mu)$ lies in the strictly feasible set $\{x \mid c(x) > 0\}$ (where no constraints are active), we can in principle search for it by using any of the unconstrained minimization algorithms described in the first part of this book. These methods need to be modified, as explained in the discussion following equation (19.30), so that they reject steps that leave the feasible region or are too close to the constraint boundaries.

One way to obtain an estimate of the Lagrange multipliers is based on differentiating P to obtain

$$\nabla_x P(x; \mu) = \nabla f(x) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x)} \nabla c_i(x). \tag{19.46}$$

When x is close to the minimizer $x(\mu)$ and μ is small, we see from Theorem 12.1 that the optimal Lagrange multipliers z_i^*, $i \in \mathcal{I}$, can be estimated as follows:

$$z_i^* \approx \mu / c_i(x), \quad i \in \mathcal{I}. \tag{19.47}$$

A general framework for algorithms based on the primal log-barrier function (19.44) can be specified as follows.

Framework 19.5 (Unconstrained Primal Barrier Method).
Given $\mu_0 > 0$, a sequence $\{\tau_k\}$ with $\tau_k \to 0$, and a starting point x_0^s;
for $k = 0, 1, 2, \ldots$
  Find an approximate minimizer x_k of $P(\cdot; \mu_k)$, starting at x_k^s,
    and terminating when $\|\nabla P(x_k; \mu_k)\| \leq \tau_k$;
    Compute Lagrange multipliers z_k by (19.47);
  if final convergence test satisfied
    stop with approximate solution x_k;
  Choose new penalty parameter $\mu_{k+1} < \mu_k$;
  Choose new starting point x_{k+1}^s;
end (**for**)

The primal barrier approach was first proposed by Frisch [115] in the 1950s and was analyzed and popularized by Fiacco and McCormick [98] in the late 1960s. It fell out of favor after the introduction of SQP methods and has not regained its popularity because it suffers from several drawbacks compared to primal-dual interior-point methods. The most

important drawback is that the minimizer $x(\mu)$ becomes more and more difficult to find as $\mu \downarrow 0$ because of the nonlinearity of the function $P(x; \mu)$

❑ EXAMPLE 19.1

Consider the problem

$$\min (x_1 + 0.5)^2 + (x_2 - 0.5)^2 \quad \text{subject to} \quad x_1 \in [0, 1], \quad x_2 \in [0, 1], \tag{19.48}$$

for which the primal barrier function is

$$\begin{aligned} P(x; \mu) = (x_1 + 0.5)^2 + (x_2 - 0.5)^2 \\ - \mu \left[\log x_1 + \log(1 - x_1) + \log x_2 + \log(1 - x_2) \right]. \end{aligned} \tag{19.49}$$

Contours of this function for the value $\mu = 0.01$ are plotted in Figure 19.1. The elongated nature of the contours indicates bad scaling, which causes poor performance of unconstrained optimization methods such as quasi-Newton, steepest descent, and conjugate gradient. Newton's method is insensitive to the poor scaling, but the nonelliptical property—the contours in Figure 19.1 are almost straight along the left edge while being circular along the right edge—indicates that the quadratic approximation on which Newton's method is based does not capture well the behavior of the barrier function. Hence, Newton's method, too, may not show rapid convergence to the minimizer of (19.49) except in a small neighborhood of this point.

❐

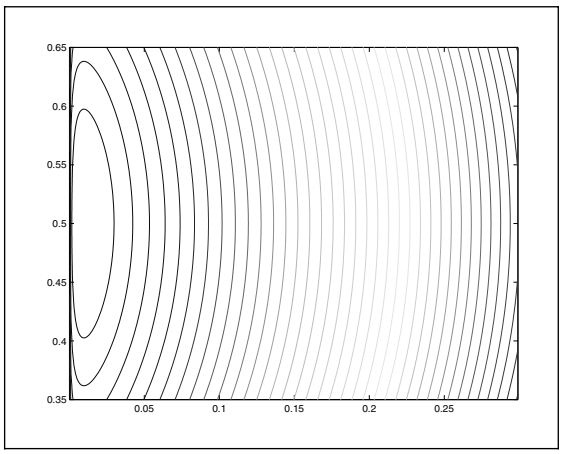

Figure 19.1
Contours of $P(x; \mu)$ from (19.49) for $\mu = 0.01$

To lessen this nonlinearity, we can proceed as in (17.21) and introduce additional variables. Defining $z_i = \mu/c_i(x)$, we rewrite the stationarity condition (19.46) as

$$\nabla f(x) - \sum_{i \in \mathcal{I}} z_i \nabla c_i(x) = 0, \tag{19.50a}$$

$$C(x)z - \mu e = 0, \tag{19.50b}$$

where $C(x) = \operatorname{diag}(c_1(x), c_2(x), \ldots, c_m(x))$. Note that this system is equivalent to the perturbed KKT conditions (19.2) for problem (19.43) if, in addition, we introduce slacks as in (19.2d). Finally, if we apply Newton's method in the variables (x, s, z) and temporarily ignore the bounds $s, z \geq 0$, we arrive at the primal-dual formulation. Thus, with hindsight, we can transform the primal log-barrier approach into the primal-dual line search approach of Section 19.4 or into the trust-region algorithm of Section 19.5.

Other drawbacks of the classical primal barrier approach are that it requires a feasible initial point, which can be difficult to find in many cases, and that the incorporation of equality constraints in a primal function is problematic. (A formulation in which the equality constraints are replaced by quadratic penalties suffers from the shortcomings of quadratic penalty functions discussed in Section 17.1.)

The shortcomings of the primal barrier approach were attributed for many years to the ill conditioning of the Hessian of the barrier function P. Note that

$$\nabla^2_{xx} P(x; \mu) = \nabla^2 f(x) - \sum_{i \in \mathcal{I}} \frac{\mu}{c_i(x)} \nabla^2 c_i(x) + \sum_{i \in \mathcal{I}} \frac{\mu}{c_i^2(x)} \nabla c_i(x) \nabla c_i(x)^T. \tag{19.51}$$

By substituting (19.47) into (19.51) and using the definition (12.33) of the Lagrangian $\mathcal{L}(x, z)$, we find that

$$\nabla^2_{xx} P(x; \mu) \approx \nabla^2_{xx} \mathcal{L}(x, z^*) + \sum_{i \in \mathcal{I}} \frac{1}{\mu} (z_i^*)^2 \nabla c_i(x) \nabla c_i(x)^T. \tag{19.52}$$

Note the similarity of this expression to the Hessian of the quadratic penalty function (17.19). Analysis of the matrix $\nabla^2_{xx} P(x; \mu)$ shows that it becomes increasingly ill conditioned near the minimizer $x(\mu)$, as μ approaches zero.

This ill conditioning will be detrimental to the performance of the steepest descent, conjugate gradient, or quasi-Newton methods. It is therefore correct to identify ill conditioning as a source of the difficulties of unconstrained primal barrier functions that use these unconstrained methods. Newton's method is, however, not affected by ill conditioning, but its performance is still not satisfactory. As explained above, it is the high *nonlinearity* of the primal barrier function P that poses significant difficulties to Newton's method.

19.7 GLOBAL CONVERGENCE PROPERTIES

We now study some global convergence properties of the primal-dual interior-point methods described in Sections 19.4 and 19.5. Theorem 19.1 provides the starting point for the analysis. It gives conditions under which limit points of the iterates generated by the interior-point methods are KKT points for the nonlinear problem. Theorem 19.1 relies on the assumption that the perturbed KKT conditions (19.2) can be satisfied (to a certain accuracy) for every value of μ_k. In this section we study conditions under which this assumption holds, that is, conditions that guarantee that our algorithms can find stationary points of the barrier problem (19.4).

We begin with a surprising observation. Whereas the line search primal-dual approach is the basis of globally convergent interior-point algorithms for linear and quadratic programming, it is not guaranteed to be successful for nonlinear programming, even for nondegenerate problems.

FAILURE OF THE LINE SEARCH APPROACH

We have seen in Chapter 11 that line search Newton iterations for nonlinear equations can fail when the Jacobian loses rank. We now discuss a different kind of failure specific to interior-point methods. It is caused by the lack of coordination between the step computation and the imposition of the bounds.

❑ **EXAMPLE 19.2** (WÄCHTER AND BIEGLER [299])

Consider the problem

$$\begin{aligned} & & \min\ & x & (19.53a)\\ \text{subject to}\quad & c_1(x) - s \stackrel{\text{def}}{=} x^2 - s_1 - 1 = 0, & (19.53b)\\ & c_2(x) - s \stackrel{\text{def}}{=} x - s_2 - \tfrac{1}{2} = 0, & (19.53c)\\ & s_1 \geq 0,\ s_2 \geq 0. & (19.53d) \end{aligned}$$

Note that the Jacobian of the equality constraints (19.53b)–(19.53c) with respect to (x, s) has full rank everywhere. Let us apply a line search interior-point method of the form (19.6)–(19.9), starting from an initial point $x^{(0)}$ such that $(s_1^{(0)}, s_2^{(0)}) > 0$, and $c_1(x^{(0)}) - s^{(0)} \geq 0$. (In this example, we use superscripts to denote iteration indices.) Figure 19.2 illustrates the feasible region (the dotted segment of the parabola) and the initial point, all projected onto the x-s_1 plane. The primal-dual step, which satisfies the linearization of the constraints (19.53b)–(19.53c), leads from $x^{(0)}$ to the tangent to the parabola. Here p_1 and p_2 are examples of possible steps satisfying the linearization of (19.53b)–(19.53c). The new iterate $x^{(1)}$ therefore lies between $x^{(0)}$ and this tangent, but since s_1 must remain positive, $x^{(1)}$ will

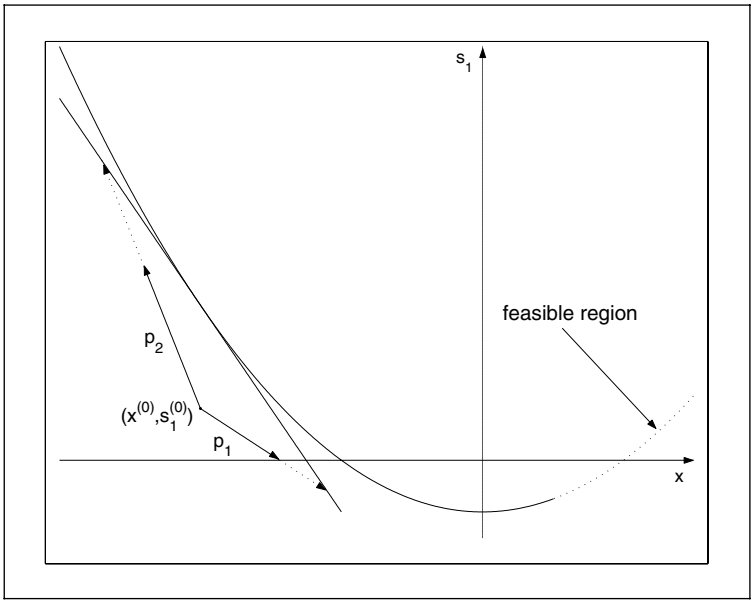

Figure 19.2 Problem (19.53) projected onto the x-s_1 plane.

lie above the horizontal axis. Thus, from any starting point above the x-axis and to the left of the parabola, namely, in the region

$$\{(x, s_1, s_2) \ : \ x^2 - s_1 - 1 \geq 0, \ s_1 \geq 0\}, \tag{19.54}$$

the new iterate will remain in this region. The argument can now be repeated to show that the iterates $\{x^{(k)}\}$ never leave the region (19.54) and therefore never become feasible.

This convergence failure affects any method that generates directions that satisfy the linearization of the constraints (19.53b)–(19.53c) and that enforces the bounds (19.53d) by the fraction to the boundary rule (19.8). The merit function can only restrict the step length further and is therefore incapable of resolving the difficulties. The strategy for updating μ is also irrelevant because the argument given above makes use only of the linearizations of the constraints.

❐

These difficulties can be observed when practical line-search codes are applied to the problem (19.53). For a wide range of starting points in the region (19.54), the interior-point iteration converges to points of the form $(-\beta, 0, 0)$, with $\beta > 0$. In other words, the iterates can converge to an infeasible, non-optimal point on the boundary of the set $\{(x_1, s_1, s_2) \ : \ s_1 \geq 0, \ s_2 \geq 0\}$, a situation that barrier methods are supposed to prevent. Furthermore, such limit points are not stationary for a feasibility measure (see Definition 17.1).

Failures of this type are rare in practice, but they highlight a theoretical deficiency of the algorithmic class (19.6)–(19.9) that may manifest itself more often as inefficient behavior than as outright convergence failure.

MODIFIED LINE SEARCH METHODS

To remedy this problem, as well as the inefficiencies caused by Hessian and constraint Jacobian singularities, we must modify the search direction of the line search interior-point iteration in some circumstances. One option is to use penalizations of the constraints [147]. Such penalty-barrier methods have been investigated only recently and mature implementations have not yet emerged.

An approach that has been successful in practice is to monitor the step lengths α_s, α_z in (19.28); if they are smaller than a given threshold, then we replace the primal-dual step by a step that guarantees progress in feasibility and, preferably, improvement in optimality, too. In a filter method, when the step lengths are very small, we can invoke the feasibility restoration phase (see Section 15.4), which is designed to generate a new iterate that reduces the infeasibility. A different approach, which assumes that a trust-region algorithm is at hand, is to replace the primal-dual step by a trust-region step, such as that produced by Algorithm 19.4.

Safeguarding the primal-dual step when the step lengths are very small is justified theoretically because, when line search iterations converge to non-stationary points, the step lengths α_s, α_z converge to zero. From a practical perspective, however, this strategy is not totally satisfactory because it attempts to react when bad steps are generated, rather than trying to prevent them. It also requires the choice of a heuristic to determine when a step length is too small. As we discuss next, the trust-region approach always generates productive steps and needs no safeguarding.

GLOBAL CONVERGENCE OF THE TRUST-REGION APPROACH

The interior-point trust-region method specified in Algorithm 19.4 has favorable global convergence properties, which we now discuss. For simplicity, we present the analysis in the context of inequality-constrained problems of the form (19.43). We first study the solution of the barrier problem (19.4) for a fixed value of μ, and then consider the complete algorithm.

In the result that follows, B_k denotes the Hessian $\nabla^2_{xx}\mathcal{L}_k$ or a quasi-Newton approximation to it. We use the measure of infeasibility $h(x) = \|[c(x)]^-\|$, where $[y] = \max\{0, -y\}$. This measure vanishes if and only if x is feasible for problem (19.43). Note that $h(x)^2$ is differentiable and its gradient is

$$\nabla[h(x)^2] = 2A(x)c(x)^-.$$

We say that a sequence $\{x_k\}$ is *asymptotically feasible* if $c(x_k)^- \to 0$. To apply Algorithm 19.4 to a fixed barrier problem, we dispense with the outer "repeat" loop.

Theorem 19.2.

Suppose that Algorithm 19.4 is applied to the barrier problem (19.4), that is, μ is fixed and the inner "repeat" loop is executed with $\epsilon_\mu = 0$. Suppose that the sequence $\{f_k\}$ is bounded below and the sequences $\{\nabla f_k\}$, $\{c_k\}$, $\{A_k\}$, and $\{B_k\}$ are bounded. Then one of the following three situations occurs:

(i) *The sequence $\{x_k\}$ is not asymptotically feasible. In this case, the iterates approach stationarity of the measure of infeasibility $h(x) = \|c(x)^-\|$, meaning that $A_k c_k^- \to 0$, and the penalty parameters ν_k tend to infinity.*

(ii) *The sequence $\{x_k\}$ is asymptotically feasible, but the sequence $\{(c_k, A_k)\}$ has a limit point $(\bar{\gamma}, \bar{A})$ failing the linear independence constraint qualification. In this situation also, the penalty parameters ν_k tend to infinity.*

(iii) *The sequence $\{x_k\}$ is asymptotically feasible, and all limit points of the sequence $\{(c_k, A_k)\}$ satisfy the linear independence constraint qualification. In this case, the penalty parameter ν_k is constant and $c_k > 0$ for all large indices k, and the stationarity conditions of problem (19.4) are satisfied in the limit.*

This theorem is proved in [48], where it is assumed, for simplicity, that Σ is given by the primal choice (19.14). The theorem accounts for two situations in which the KKT conditions may not be satisfied in the limit, both of which are of interest. Outcome (i) is a case in which, in the limit, there is no direction that improves feasibility to first order. This outcome cannot be ruled out because finding a feasible point is a problem that a local method cannot always solve without a good starting point. (Note that we do not assume that the constraint Jacobian A_k has full rank.)

In considering outcome (ii), we must keep in mind that in some cases the solution to problem (19.43) is a point where the linear independence constraint qualification fails and that is not a KKT point. Outcome (iii) is the most desirable outcome and can be monitored in practice by observing, for example, the behavior of the penalty parameter ν_k.

We now study the complete interior-point method given in Algorithm 19.4 applied to the nonlinear programming problem (19.43). By combining Theorems 19.1 and 19.2 we see that the following outcomes can occur:

- For some barrier parameter μ generated by the algorithm, either the inequality $\|c_k - s_k\| \leq \epsilon_\mu$ is never satisfied, in which case the stationarity condition for minimizing $h(x)$ is satisfied in the limit, or else $(c_k - s_k) \to 0$, in which case the sequence $\{(c_k, A_k)\}$ has a limit point $(\bar{c}, \bar{A})$ failing the linear independence constraint qualification;

- At each outer iteration of Algorithm 19.4 the inner stop test $E(x_k, s_k, y_k, z_k; \mu) \leq \epsilon_\mu$ is satisfied. Then all limit points of the iteration sequence are feasible. Furthermore,

if any limit point $\hat{x}$ satisfies the linear independence constraint qualification, the first-order necessary conditions for problem (19.43) hold at $\hat{x}$.

19.8 SUPERLINEAR CONVERGENCE

We can implement primal-dual interior-point methods so that they converge quickly near the solution. All is needed is that we carefully control the decrease in the barrier parameter μ and the inner convergence tolerance ϵ_μ, and let the parameter τ in (19.9) converge to 1 sufficiently rapidly. We now describe strategies for updating these parameters in the context of the line search iteration discussed in Section 19.4; these strategies extend easily to the trust-region method of Section 19.5.

In the discussion that follows, we assume that the merit function or filter is inactive. This assumption is realistic because with a careful implementation (which may include second-order correction steps or other features), we can ensure that, near a solution, all the steps generated by the primal-dual method are acceptable to the merit function or filter.

We denote the primal-dual iterates by

$$v = (x, s, y, z) \tag{19.55}$$

and define the full primal-dual step (without backtracking) by

$$v^+ = v + p, \tag{19.56}$$

where p is the solution of (19.12). To establish local convergence results, we assume that the iterates converge to a solution point satisfying certain regularity assumptions.

Assumptions 19.1.

(a) *v^* is a solution of the nonlinear program (19.1) for which the first-order KKT conditions are satisfied.*

(b) *The Hessian matrices $\nabla^2 f(x)$ and $\nabla^2 c_i(x)$, $i \in \mathcal{E} \cup \mathcal{I}$, are locally Lipschitz continuous at v^*.*

(c) *The linear independence constraint qualification (LICQ) (Definition 12.4), the strict complementarity condition (Definition 12.5), and the second-order sufficient conditions (Theorem 12.6) hold at v^*.*

We assume that v is an iterate at which the inner stop test $E(v, \mu) \leq \epsilon_\mu$ is satisfied, so that the barrier parameter is decreased from μ to μ^+. We now study how to control the parameters in Algorithm 19.2 so that the following three properties hold in a neighborhood of v^*:

1. The iterate v^+ satisfies the fraction to the boundary rule (19.9), that is, $\alpha_s^{\max} = \alpha_z^{\max} = 1$.
2. The inner stop test is satisfied at v^+, that is, $E(v^+; \mu^+) \leq \epsilon_{\mu^+}$.
3. The sequence of iterates (19.56) converge superlinearly to v^*.

We can achieve these three goals by letting

$$\epsilon_\mu = \theta\mu \qquad \text{and} \qquad \epsilon_{\mu^+} = \theta\mu^+, \tag{19.57}$$

for $\theta > 0$, and setting the other parameters as follows:

$$\mu^+ = \mu^{1+\delta}, \quad \delta \in (0, 1); \qquad \tau = 1 - \mu^\beta, \quad \beta > \delta. \tag{19.58}$$

There are other practical ways of controlling the parameters of the algorithm. For example, we may prefer to determine the change in μ from the reduction achieved in the KKT conditions of the nonlinear program, as measured by the function E. The three results mentioned above can be established if the convergence tolerance ϵ_μ is defined as in (19.57) and if we replace μ by $E(v; 0)$ in the right-hand sides of the definitions (19.58) of μ^+ and τ.

There is a limit to how fast we can decrease μ and still be able to satisfy the inner stop test after just one iteration (condition 2). One can show that there is no point in decreasing μ at a faster than quadratic rate, since the overall convergence cannot be faster than quadratic. Not suprising, if τ is constant and $\mu^+ = \sigma\mu$, with $\sigma \in (0, 1)$, then the interior-point algorithm is only linearly convergent.

Although it is desirable to implement interior-point methods so that they achieve a superlinear rate of convergence, this rate is typically observed only in the last few iterations in practice.

19.9 PERSPECTIVES AND SOFTWARE

Software packages that implement nonlinear interior-point methods are widely available. Line search implementations include LOQO [294], KNITRO/DIRECT [303], IPOPT [301], and BARNLP [21], and for convex problems, MOSEK [5]. The trust-region algorithm discussed in Section 19.5 has been implemented in KNITRO/CG [50]. These interior-point packages have proved to be strong competitors of the leading active-set and augmented Lagrangian packages, such as MINOS [218], SNOPT [128], LANCELOT [72], FILTERSQP [105], and KNITRO/ACTIVE [49]. At present, interior-point and active-set methods appear to be the most promising approaches, while augmented Lagrangian methods seem to be less efficient. The KNITRO package provides crossover from interior-point to active-set modes [46].

Interior-point methods show their strength in large-scale applications, where they often (but not always) outperform active-set methods. In interior-point methods, the linear

system to be solved at every iteration has the same block structure, so effort can be focused on exploiting this structure. Both direct factorization techniques and projected CG methods are available, allowing the user to solve many types of applications efficiently. On the other hand, interior-point methods, unlike active-set methods, consider all the constraints at each iteration, even if they are irrelevant to the solution. As a result, the cost of the primal-dual iteration can be excessive in some applications.

One of the main weaknesses of interior-point methods is their sensitivity to the choice of the initial point, the scaling of the problem, and the update strategy for the barrier parameter μ. If the iterates approach the boundary of the feasible region prematurely, interior-point methods may have difficulty escaping it, and convergence can be slow. The availability of adaptive strategies for updating μ is, however, beginning to lessen this sensitivity, and more robust implementations can be expected in the coming years.

Although the description of the line search algorithm in Section 19.4 is fairly complete, various details of implementation (such as second-order corrections, iterative refinement, and resetting of parameters) are needed to obtain a robust code. Our description of the trust-region method of Algorithm 19.4 leaves some important details unspecified, particularly concerning the procedure for computing approximate solutions of the normal and tangential subproblems; see [50] for further discussion. The KNITRO/CG implementation of this trust-region algorithm uses a projected CG iteration in the computation of the step, which allows the method to work even when only Hessian–vector products are available, not the Hessian itself.

Filters and merit functions have each been used to globalize interior-point methods. Although some studies have shown that merit functions restrict the progress of the iteration unduly [298], recent developments in penalty update procedures (see Chapter 18) have altered the picture, and it is currently unclear whether filter globalization approaches are preferable.

NOTES AND REFERENCES

The development of modern nonlinear interior-point methods was influenced by the success of interior-point methods for linear and quadratic programming. The concept of primal-dual steps arises from the homotopy formulation given in Section 19.1, which is an extension of the systems (14.13) and (16.57) for linear and quadratic programming. Although the primal barrier methods of Section 19.6 predate primal-dual methods by at least 15 years, they played a limited role in their development.

There is a vast literature on nonlinear interior-point methods. We refer the reader to the surveys by Forsgren, Gill, and Wright [111] and Gould, Orban, and Toint [147] for a comprehensive list of references. The latter paper also compares and contrasts interior-point methods with other nonlinear optimization methods. For an analysis of interior-point methods that use filter globalization see, for example, Ulbrich, Ulbrich, and Vicente [291] and Wächter and Biegler [300]. The book by Conn, Gould, and Toint [74] gives a thorough presentation of several interior-point methods.

Primal barrier methods were originally proposed by Frisch [115] and were analyzed in an authoritative book by Fiacco and McCormick [98]. The term "interior-point method" and the concept of the primal central path $\mathcal{C}_p$ appear to have originated in this book. Nesterov and Nemirovskii [226] propose and analyze several families of barrier methods and establish polynomial-time complexity results for very general classes of problems such as semidefinite and second-order cone programming. For a discussion of the history of barrier function methods, see Nash [221].

EXERCISES

✎ **19.1** Consider the nonlinear program

$$\min \ f(x) \ \text{ subject to } \ c_E(x) = 0, \quad c_I(x) \geq 0. \tag{19.59}$$

(a) Write down the KKT conditions of (19.1) and (19.59), and establish a one-to-one correspondence between KKT points of these problems (despite the different numbers of variables and multipliers).

(b) The multipliers z correspond to the *equality* constraints (19.1c) and should therefore be unsigned. Nonetheless, argue that (19.2) with $\mu = 0$ together with (19.3) can be seen as the KKT conditions of problem (19.1). Moreover, argue that the multipliers z in (19.2) can be seen as the multipliers of the inequalities c_I in (19.59).

(c) Suppose $\bar{x}$ is feasible for (19.59). Show that LICQ holds at $\bar{x}$ for (19.59) if and only if LICQ holds at $(\bar{x}, \bar{s})$ for (19.1), with $\bar{s} = c_I(\bar{x})$.

(d) Repeat part (c) assuming that the MFCQ condition holds (see Definition 12.6) instead of LICQ.

✎ **19.2** This question concerns Algorithm 19.1.

(a) Extend the proof of Theorem 19.1 to the general nonlinear program (19.1).

(b) Show that the theorem still holds if the condition $E(x_k, s_k, y_k, z_k) \leq \mu_k$ is replaced by $E(x_k, s_k, y_k, z_k) \leq \epsilon_{\mu_k}$, for any sequence ϵ_{μ_k} that converges to 0 as $\mu_k \to 0$.

(c) Suppose that in Algorithm 19.1 the new iterate $(x_{k+1}, s_{k+1}, y_{k+1}, z_{k+1})$ is obtained by *any* means. What conditions are required on this iterate so that Theorem 19.1 holds?

✎ **19.3** Consider the nonlinear system of equations (11.1). Show that Newton's method (11.6) is invariant to scalings of the equations. More precisely, show that the Newton step p does not change if each component of r is multiplied by a nonzero constant.