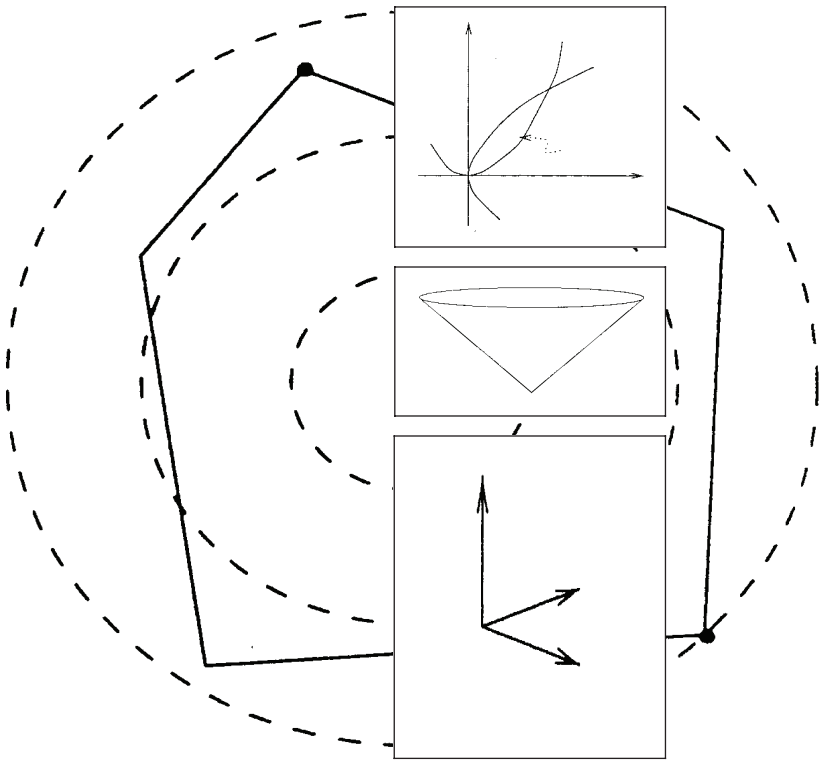
CHAPTER 16



Quadratic Programming

An optimization problem with a quadratic objective function and linear constraints is called a quadratic program. Problems of this type are important in their own right, and they also arise as subproblems in methods for general constrained optimization, such as sequential quadratic programming (Chapter 18) and augmented Lagrangian methods (Chapter 17).

The general quadratic program (QP) can be stated as

$$\min_x \; q(x) = \tfrac{1}{2} x^T G x + x^T d \tag{16.1a}$$

$$\text{subject to } a_i^T x = b_i, \qquad i \in \mathcal{E}, \tag{16.1b}$$

$$a_i^T x \geq b_i, \qquad i \in \mathcal{I}, \tag{16.1c}$$

where G is a symmetric $n \times n$ matrix, $\mathcal{E}$ and $\mathcal{I}$ are finite sets of indices, and d, x, and $\{a_i\}$, $i \in \mathcal{E} \cup \mathcal{I}$, are vectors with n elements. Quadratic programs can always be solved (or can be shown to be infeasible) in a finite number of iterations, but the effort required to find a solution depends strongly on the characteristics of the objective function and the number of inequality constraints. If the Hessian matrix G is positive semidefinite, we say that (16.1) is a *convex QP*, and in this case the problem is sometimes not much more difficult to solve

than a linear program. *Nonconvex* QPs, in which G is an indefinite matrix, can be more challenging, since they can have several stationary points and local minima.

In this chapter we limit ourselves to studying algorithms that find the solution of a convex quadratic program or a stationary point of a general (nonconvex) quadratic program. We start by considering an interesting application of quadratic programming.

AN EXAMPLE: PORTFOLIO OPTIMIZATION

Every investor knows that there is a tradeoff between risk and return: To increase the expected return on investment, an investor must be willing to tolerate greater risks. Portfolio theory studies how to model this tradeoff given a collection of n possible investments with returns r_i, $i = 1, 2, \ldots, n$. The returns r_i are usually not known in advance, and are often assumed to be random variables that follow a normal distribution. We can characterize these variables by their expected value $\mu_i = E[r_i]$ and their variance $\sigma_i^2 = E[(r_i - \mu_i)^2]$. The variance measures the fluctuations of the variable r_i about its mean, so that larger values of σ_i indicate riskier investments.

An investor constructs a portfolio by putting a fraction x_i of the available funds into investment i, for $i = 1, 2, \ldots, n$. Assuming that all available funds are invested and that short-selling is not allowed, the constraints are $\sum_{i=1}^{n} x_i = 1$ and $x \geq 0$. The return on the portfolio is given by

$$R = \sum_{i=1}^{n} x_i r_i. \tag{16.2}$$

To measure the desirability of the portfolio, we need to obtain measures of its expected return variance. The expected return is simply

$$E(R) = E\left[\sum_{i=1}^{n} x_i r_i\right] = \sum_{i=1}^{n} x_i E[r_i] = x^T \mu.$$

The variance, too, can be calculated from elementary laws of statistics. It depends on the *covariances* between each pair of investments, which are defined by

$$\rho_{ij} = \frac{E[(r_i - \mu_i)(r_j - \mu_j)]}{\sigma_i \sigma_j}, \qquad \text{for } i, j = 1, 2, \ldots, n.$$

The correlation measures the tendency of the return on investments i and j to move in the same direction. Two investments whose returns tend to rise and fall together have a positive covariance; the nearer ρ_{ij} is to 1, the more closely the two investments track each other. Investments whose returns tend to move in opposite directions have negative covariance.

The variance of the total portfolio R is then given by

$$E[(R - E[R])^2] = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \sigma_i \sigma_j \rho_{ij} = x^T G x,$$

where we have defined the $n \times n$ symmetric matrix G by

$$G_{ij} = \rho_{ij} \sigma_i \sigma_j.$$

It can be shown that G is positive semidefinite.

We are interested in portfolios for which the expected return $x^T \mu$ is large while the variance $x^T G x$ is small. In the model proposed by Markowitz [157], we combine these two aims into a single objective function with the aid of a "risk tolerance parameter" denoted by κ, and solve the following problem to find the "optimal" portfolio:

$$\max \; x^T \mu - \kappa x^T G x, \quad \text{subject to} \quad \sum_{i=1}^{n} x_i = 1, \; x \geq 0.$$

The parameter κ lies in the range $[0, \infty)$, and its chosen value depends on the preferences of the individual investor. Conservative investors would place more emphasis on minimizing risk in their portfolio, so they would choose a large value of κ to increase the weight of the variance measure in the objective function. More daring investors are prepared to take on more risk in the hope of a higher expected return, so their value of κ would be closer to zero.

The difficulty in applying this portfolio optimization technique to real-life investing lies in defining the expected returns, variances, and covariances for the investments in question. One possibility is to use historical data, defining the quantities μ_i, σ_i, and ρ_{ij} to be equal to their historical values between the present day and, say, five years ago. It is not wise to assume that future performance will mirror the past, of course. Moreover, historical data will not be available for many interesting investments (such as start-up companies based on new technology). Financial professionals often combine the historical data with their own insights and expectations to produce values of μ_i, σ_i, and ρ_{ij}.

16.1 EQUALITY--CONSTRAINED QUADRATIC PROGRAMS

We begin our discussion of algorithms for quadratic programming by considering the case where only equality constraints are present. As we will see in this chapter, active set methods for general quadratic programming solve an equality–constrained QP at each iteration.

PROPERTIES OF EQUALITY-CONSTRAINED QPS

Let us denote the number of constraints by m, assume that $m \leq n$, and write the quadratic program as

$$\min_x \; q(x) \stackrel{\text{def}}{=} \tfrac{1}{2} x^T G x + x^T d \tag{16.3a}$$

$$\text{subject to} \quad Ax = b, \tag{16.3b}$$

where A is the $m \times n$ Jacobian of constraints defined by

$$A = [a_i]_{i \in \mathcal{E}}^T.$$

For the present, we assume that A has full row rank (rank m), and that the constraints (16.3b) are consistent. (Toward the end of the chapter we discuss the case in which A is rank deficient.)

The first-order necessary conditions for x^* to be a solution of (16.3) state that there is a vector λ^* such that the following system of equations is satisfied:

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -d \\ b \end{bmatrix}. \tag{16.4}$$

It is easy to derive these conditions as a consequence of the general result for first-order optimality conditions, Theorem 12.1. As in Chapter 12, we call λ^* the vector of Lagrange multipliers. The system (16.4) can be rewritten in a form that is useful for computation by expressing x^* as $x^* = x + p$, where x is some estimate of the solution and p is the desired step. By introducing this notation and rearranging the equations, we obtain

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ c \end{bmatrix}, \tag{16.5}$$

where

$$c = Ax - b, \qquad g = d + Gx, \qquad p = x^* - x. \tag{16.6}$$

The matrix in (16.5) is called the Karush–Kuhn–Tucker (KKT) matrix, and the following result gives conditions under which it is nonsingular. As in Chapter 15, we use Z to denote the $n \times (n - m)$ matrix whose columns are a basis for the null space of A. That is, Z has full rank and $AZ = 0$.

Lemma 16.1.

Let A have full row rank, and assume that the reduced-Hessian matrix $Z^T G Z$ is positive definite. Then the KKT matrix

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \tag{16.7}$$

is nonsingular, and there is a unique vector pair (x^*, λ^*) satisfying (16.4).

PROOF. Suppose there are vectors p and v such that

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = 0. \tag{16.8}$$

Since $Ap = 0$, we have from (16.8) that

$$0 = \begin{bmatrix} p \\ v \end{bmatrix}^T \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} = p^T G p.$$

Since p lies in the null space of A, it can be written as $p = Zu$ for some vector $u \in \mathbb{R}^{n-m}$. Therefore, we have

$$0 = p^T G p = u^T Z^T G Z u,$$

which by positive definiteness of $Z^T G Z$ implies that $u = 0$. Therefore, $p = 0$, and by (16.8), $A^T v = 0$; and the full row rank of A implies that $v = 0$. We conclude that equation (16.8) is satisfied only if $p = 0$ and $v = 0$, so the matrix is nonsingular, as claimed. □

❑ EXAMPLE 16.1

Consider the quadratic programming problem

$$\begin{aligned} \min\ & q(x) = 3x_1^2 + 2x_1x_2 + x_1x_3 + 2.5x_2^2 + 2x_2x_3 + 2x_3^2 - 8x_1 - 3x_2 - 3x_3, \\ & \text{subject to} \quad x_1 + x_3 = 3, \quad x_2 + x_3 = 0. \end{aligned} \tag{16.9}$$

We can write this problem in the form (16.3) by defining

$$G = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}, \quad d = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

The solution x^* and optimal Lagrange multiplier vector λ^* are given by

$$x^* = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \quad \lambda^* = \begin{bmatrix} 3 \\ -2 \end{bmatrix}.$$

In this example, the matrix G is positive definite, and the null-space basis matrix can be defined as

$$Z = (-1, -1, 1)^T. \tag{16.10}$$

□

We have seen that when the conditions of Lemma 16.1 are satisfied, then there is a unique vector pair (x^*, λ^*) that satisfies the first-order necessary conditions for (16.3). In fact, the second-order sufficient conditions (see Theorem 12.6) are also satisfied at (x^*, λ^*), so x^* is a strict local minimizer of (16.3). However, we can use a direct argument to show that x^* is actually a *global* solution of (16.3).

Theorem 16.2.

Suppose that the conditions of Lemma 16.1 are satisfied. Then the vector x^* satisfying (16.4) is the unique global solution of (16.3).

PROOF. Let x be any other feasible point (satisfying $Ax = b$), and as before, we use p to denote the difference $x^* - x$. Since $Ax^* = Ax = b$, we have that $Ap = 0$. By substituting into the objective function (16.3a), we obtain

$$\begin{aligned} q(x) &= \tfrac{1}{2}(x^* - p)^T G(x^* - p) + d^T(x^* - p) \\ &= \tfrac{1}{2} p^T G p - p^T G x^* - d^T p + q(x^*). \end{aligned} \tag{16.11}$$

From (16.4) we have that $Gx^* = -d + A^T\lambda^*$, so from $Ap = 0$ we have that

$$p^T G x^* = p^T(-d + A^T\lambda^*) = -p^T d.$$

By substituting this relation into (16.11), we obtain

$$q(x) = \tfrac{1}{2} p^T G p + q(x^*).$$

Since p lies in the null space of A, we can write $p = Zu$ for some vector $u \in \mathbb{R}^{n-m}$, so that

$$q(x) = \tfrac{1}{2} u^T Z^T G Z u + q(x^*).$$

By positive definiteness of $Z^T G Z$, we conclude that $q(x) > q(x^*)$ except when $u = 0$, that is, when $x = x^*$. Therefore, x^* is the unique global solution of (16.3). □

When the projected Hessian matrix $Z^T G Z$ has zero or negative eigenvalues, the problem (16.3) does not have a bounded solution, except in a special case. To demonstrate this claim, suppose that there is a vector pair (x^*, λ^*) that satisfies the KKT conditions (16.4). Let u be some vector such that $u^T Z^T G Z u \leq 0$, and set $p = Zu$. Then for any $\alpha > 0$, we have that

$$A(x^* + \alpha p) = b,$$

so that $x^* + \alpha p$ is feasible, while

$$\begin{aligned} q(x^* + \alpha p) &= q(x^*) + \alpha p^T (G x^* + d) + \tfrac{1}{2}\alpha^2 p^T G p \\ &= q(x^*) + \tfrac{1}{2}\alpha^2 p^T G p \leq q(x^*), \end{aligned}$$

where we have used the facts that $Gx^* + d = A^T \lambda^*$ from (16.4) and $p^T A^T \lambda^* = u^T Z^T A^T \lambda^* = 0$. Therefore, from any x^* satisfying the KKT conditions, we can find a feasible direction p along which q does not increase. In fact, we can always find a direction of *strict decrease* for q unless $Z^T G Z$ has no negative eigenvalues. The only case in which (16.3) has solutions is the one in which there exists some point x^* for which (16.4) is satisfied, while $Z^T G Z$ is positive semidefinite. Even in this case, the solution is not a strict local minimizer.

16.2 SOLVING THE KKT SYSTEM

In this section we discuss efficient methods for solving the KKT system (16.4) (or, alternatively, (16.5)).

The first important observation is that if $m \geq 1$, the KKT matrix is always indefinite. The following result characterizes the *inertia* of (16.7) under the assumptions of Lemma 16.1. The inertia of a matrix is the scalar triple that indicates the number of its positive, negative, and zero eigenvalues.

Lemma 16.3.

Suppose that A has full row rank and that the reduced Hessian $Z^T G Z$ is positive definite. Then the KKT matrix (16.7) has n positive eigenvalues, m negative eigenvalues, and no zero eigenvalues.

This result follows from Theorem 16.6 given later in this chapter. Knowing that the KKT system is indefinite, we now describe the main techniques developed for its solution.

DIRECT SOLUTION OF THE KKT SYSTEM

One option for solving (16.5) is to perform a triangular factorization on the full KKT matrix and then perform backward and forward substitution with the triangular factors. We cannot use the Cholesky factorization algorithm because the KKT matrix is indefinite. Instead, we could use Gaussian elimination with partial pivoting (or a sparse variant of this algorithm) to obtain the L and U factors, but this approach has the disadvantage that it ignores the symmetry of the system.

The most effective strategy in this case is to use a *symmetric indefinite factorization*. We have described these types of factorizations in Chapter 6. For a general symmetric matrix K, these factorizations have the form

$$P^T K P = L B L^T, \tag{16.12}$$

where P is a permutation matrix, L is unit lower triangular, and B is block-diagonal with either 1×1 or 2×2 blocks. The symmetric permutations defined by the matrix P are introduced for numerical stability of the computation and, in the case of large sparse K, to maintain sparsity. The computational cost of symmetric indefinite factorization (16.12) is typically about half the cost of sparse Gaussian elimination.

To solve (16.5) we first compute the factorization (16.12), substituting the KKT matrix for K. We then perform the following sequence of operations to arrive at the solution:

$$\begin{aligned}
&\text{solve } Ly = P^T \begin{bmatrix} g \\ c \end{bmatrix} \text{ to obtain } y; \\
&\text{solve } B\hat{y} = y \text{ to obtain } \hat{y}; \\
&\text{solve } L^T \bar{y} = \hat{y} \text{ to obtain } \bar{y}; \\
&\text{set } \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = P\bar{y}.
\end{aligned}$$

Note that multiplications with the permutation matrices P and P^T can be performed by rearranging vector components, and are therefore inexpensive. Solution of the system $B\hat{y} = y$ entails solving a number of small 1×1 and 2×2 systems, so the number of operations is a small multiple of the vector length $(m+n)$, again quite inexpensive. Triangular substitutions with L and L^T are more significant. Their precise cost depends on the amount of sparsity, but is usually significantly less than the cost of performing the factorization (16.12).

This approach of factoring the full $(n+m) \times (n+m)$ KKT matrix (16.7) is quite effective on some problems. Difficulties may arise when the heuristics for choosing the permutation matrix P are not able to do a very good job of maintaining sparsity in the L factor, so that L becomes much more dense than the original coefficient matrix.

An alternative to the direct factorization approach for the matrix in (16.5) is to apply an iterative method. The conjugate gradient method is not recommended because it can be

unstable on systems that are not positive definite. Therefore, we must consider techniques for general linear systems, or for symmetric indefinite systems. Candidates include the QMR and LSQR methods (see the Notes and References at the end of the chapter).

RANGE-SPACE METHOD

In the range-space method, we use the matrix G to perform block elimination on the system (16.5). Assuming that G is positive definite, we multiply the first equation in (16.5) by AG^{-1} and then subtract the second equation to obtain a linear system in the vector λ^* alone:

$$(AG^{-1}A^T)\lambda^* = (AG^{-1}g - c). \tag{16.13}$$

We solve this symmetric positive definite system for λ^*, and then recover p from the first equation in (16.5) by solving

$$Gp = A^T\lambda^* - g. \tag{16.14}$$

This approach requires us to perform operations with G^{-1}, as well as to compute the factorization of the $m \times m$ matrix $AG^{-1}A^T$. It is therefore most useful when:

- G is well conditioned and easy to invert (for instance, when G is diagonal or block-diagonal);
- G^{-1} is known explicitly through a quasi-Newton updating formula; or
- the number of equality constraints m is small, so that the number of backsolves needed to form the matrix $AG^{-1}A^T$ is not too large.

The range-space approach is, in effect, a special case of the symmetric elimination approach of the previous section. It corresponds to the case in which the first n variables in the system (16.5) are eliminated *before* we eliminate any of the last m variables. In other words, it is obtained if we choose the matrix P in (16.12) as

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix},$$

where P_1 and P_2 are permutation matrices of dimension $n \times n$ and $m \times m$, respectively.

We can actually use a similar approach to the range-space method to derive an explicit inverse formula for the KKT matrix in (16.5). This formula is

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}^{-1} = \begin{bmatrix} C & E \\ E^T & F \end{bmatrix}, \tag{16.15}$$

with

$$C = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1},$$
$$E = G^{-1}A^T(AG^{-1}A^T)^{-1},$$
$$F = -(AG^{-1}A^T)^{-1}.$$

The solution of (16.5) can be obtained by multiplying its right-hand-side by this matrix. If we take advantage of common expressions and group terms appropriately, we recover the approach (16.13), (16.14).

NULL-SPACE METHOD

The null-space method, which we now describe, does not require nonsingularity of G and is therefore of wider applicability than the range-space method. It assumes only that the conditions of Lemma 16.1 hold, namely, that A has full row rank and that Z^TGZ is positive definite. It requires, however, knowledge of the null-space basis matrix Z. Like the range-space method, it exploits the block structure in the KKT system to decouple (16.5) into two smaller systems.

Suppose that we partition the vector p in (16.5) into two components, as follows:

$$p = Yp_Y + Zp_Z, \tag{16.16}$$

where Z is the $n \times (n-m)$ null-space matrix, Y is any $n \times m$ matrix such that $[Y \mid Z]$ is nonsingular, p_Y is an m-vector, and p_Z is an $(n-m)$-vector (see Chapter 15). As illustrated in Figure 15.3, Yp_Y is a particular solution of $Ax = b$, and Zp_Z is a displacement along these constraints.

By substituting p into the second equation of (16.5) and recalling that $AZ = 0$, we obtain

$$(AY)p_Y = -c. \tag{16.17}$$

Since A has rank m and $[Y \mid Z]$ is $n \times n$ nonsingular, the product $A[Y \mid Z] = [AY \mid 0]$ has rank m. Therefore, AY is a nonsingular $m \times m$ matrix, and p_Y is well determined by the equations (16.17). Meanwhile, we can substitute (16.16) into the first equation of (16.5) to obtain

$$-GYp_Y - GZp_Z + A^T\lambda^* = g$$

and multiply by Z^T to obtain

$$(Z^TGZ)p_Z = -[Z^TGYp_Y + Z^Tg]. \tag{16.18}$$

This system, which can be solved by means of the Cholesky factorization of the $(n-m) \times (n-m)$ reduced-Hessian matrix $Z^T G Z$, determines p_Z, and hence the total displacement $p = Y p_Y + Z p_Z$. To obtain the Lagrange multiplier, we multiply the first equation of (16.5) by Y^T to obtain the linear system

$$(AY)^T \lambda^* = Y^T (g + Gp), \tag{16.19}$$

which can be solved for λ^*.

❑ EXAMPLE 16.2

Consider the problem (16.9) given in the previous example. We can choose

$$Y = \begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \\ 1/3 & 1/3 \end{pmatrix}$$

and Z is as in (16.10). For this particular choice of Y, we have $AY = I$.

Suppose we have $x = (0, 0, 0)^T$ in (16.5). Then

$$c = Ax - b = -b, \qquad g = d + Gx = d = (-8, -3, -3)^T.$$

Simple calculation shows that

$$p_Y = (3, 0)^T, \qquad p_Z = 0,$$

so that

$$p = x^* - x = Y p_Y + Z p_Z = (2, -1, 1)^T.$$

After recovering λ^* from (16.19), we conclude that

$$x^* = (2, -1, 1)^T, \qquad \lambda^* = (3, -2)^T.$$

The null-space approach can be effective when the number of degrees of freedom $n - m$ is small. Its main drawback is the need for the null-space matrix Z, which as we have seen in Chapter 15, can be expensive to compute in many large problems. The matrix Z is not uniquely defined, and if it is poorly chosen, the reduced system (16.18) may become ill conditioned. If we choose Z to have orthonormal columns, as is normally done in software for small and medium-sized problems, then the conditioning of $Z^T G Z$ is at least as good as

that of G itself. When A is large and sparse, however, this choice of Z is relatively expensive to compute, so for practical reasons we are often forced to use one of the less reliable choices of Z described in Chapter 15.

The reduced system (16.18) also can be solved by means of the conjugate gradient (CG) method. If we adopt this approach, it is not necessary to form the reduced Hessian $Z^T G Z$ explicitly, since the CG method requires only that we compute matrix–vector products involving this matrix. In fact, it is not even necessary to form Z explicitly, as long as we are able to compute products of Z and Z^T with arbitrary vectors. For some choices of Z and for large problems, these products are much cheaper to compute than Z itself, as we have seen in Chapter 15. A drawback of the conjugate gradient approach in the absence of an explicit representation of Z is that standard preconditioning techniques, such as modified Cholesky, cannot be used, and the development of effective preconditioners for this case is still the subject of investigation.

It is difficult to give hard and fast rules about the relative effectiveness of null-space and range-space methods, since factors such as fill-in during computation of Z vary significantly even among problems of the same dimension. In general, we can recommend the range-space method when G is positive definite and $AG^{-1}A^T$ can be computed cheaply (because G is easy to invert or because m is small relative to n). Otherwise, the null-space method is often preferable, in particular when it is much more expensive to compute factors of G than to compute the null-space matrix Z and the factors of $Z^T G Z$.

A METHOD BASED ON CONJUGACY

The range-space and null-space methods described above are useful in a wide variety of settings and applications. The following approach is of more limited use, but has proved to be the basis for efficient methods for convex QP. It is applicable in the case where the Hessian G is positive definite, and can be regarded as a null-space method that makes clever use of conjugacy to provide a particularly simple formula for the step computation. (See Chapter 5 for a definition of conjugacy.)

The key idea is to compute an $n \times n$ nonsingular matrix W with the following properties:

$$W^T G W = I, \qquad AW = \begin{bmatrix} 0 & U \end{bmatrix}, \tag{16.20}$$

where U is an $m \times m$ upper triangular matrix. The first equation in (16.20) states that the columns of W are conjugate with respect to the Hessian G, whereas the second condition implies that the first $n - m$ columns of W lie in the null space of A.

The matrix W can be constructed explicitly by using the QR and Cholesky factorizations. (These factorizations are described in Section A.2.) By using a variant of the QR factorization (see Exercise 6), we can find an orthogonal matrix Q and an $m \times m$ upper triangular matrix $\hat{U}$ such that

$$AQ = \begin{bmatrix} 0 & \hat{U} \end{bmatrix}.$$

Since $Q^T G Q$ is symmetric and positive definite, we can compute its Cholesky decomposition $Q^T G Q = L L^T$. By defining $W = Q L^{-T}$, we find that (16.20) is satisfied with $U = \hat{U} L^{-T}$.

We can partition the columns of W to obtain the matrices Z and Y used in (16.16), by writing

$$W = \begin{bmatrix} Z & Y \end{bmatrix},$$

where Z is the first $n - m$ columns and Y is the last m columns. By using this definition and (16.20), we find that

$$Z^T G Z = I, \quad Z^T G Y = 0, \quad Y^T G Y = I, \tag{16.21}$$

and

$$AY = U, \quad AZ = 0. \tag{16.22}$$

These relations allow us to simplify the equations (16.17), (16.18), and (16.19) that are used to solve for p_Y, p_Z, and λ^*, respectively. We have

$$U p_Y = -c, \tag{16.23a}$$
$$p_Z = -Z^T g, \tag{16.23b}$$
$$U^T \lambda^* = Y^T g + p_Y, \tag{16.23c}$$

so these vectors can be recovered at the cost of two triangular substitutions involving U and matrix–vector products involving Y^T and Z^T. The vector p can then be recovered from (16.16).

16.3 INEQUALITY-CONSTRAINED PROBLEMS

In the remainder of the chapter we discuss several classes of algorithms for solving QPs that contain inequality constraints and possibly equality constraints. *Classical active-set methods* can be applied both to convex and nonconvex problems, and they have been the most widely used methods since the 1970s. *Gradient–projection methods* attempt to accelerate the solution process by allowing rapid changes in the active set, and are most efficient when the only constraints in the problem are bounds on the variables. *Interior-point methods* have recently been shown to be effective for solving large convex quadratic programs.

We discuss these methods in the following sections. We mention also that quadratic programs can also be solved by the augmented Lagrangian methods of Chapter 17, or by means of an exact penalization method such as the Sℓ_1QP approach discussed in Chapter 18.