INERTIA-CONTROLLING METHODS FOR GENERAL QUADRATIC PROGRAMMING*

PHILIP E. GILL†, WALTER MURRAY‡, MICHAEL A. SAUNDERS‡, AND MARGARET H. WRIGHT§

Abstract. Active-set quadratic programming (QP) methods use a working set to define the search direction and multiplier estimates. In the method proposed by Fletcher in 1971, and in several subsequent mathematically equivalent methods, the working set is chosen to control the inertia of the reduced Hessian, which is never permitted to have more than one nonpositive eigenvalue. (We call such methods *inertia-controlling.*) This paper presents an overview of a generic inertia-controlling QP method, including the equations satisfied by the search direction when the reduced Hessian is positive definite, singular and indefinite. Recurrence relations are derived that define the search direction and Lagrange multiplier vector through equations related to the Karush-Kuhn-Tucker system. We also discuss connections with inertia-controlling methods that maintain an explicit factorization of the reduced Hessian matrix.

Key words. Nonconvex quadratic programming, active-set methods, Schur complement, Karush-Kuhn-Tucker system, primal-feasible methods.

1. Introduction. The *quadratic programming* (QP) problem is to minimize a quadratic objective function subject to linear constraints on the variables. The linear constraints may include an arbitrary mixture of equality and inequality constraints, where the latter may be subject to lower and/or upper bounds. Many mathematically equivalent formulations are possible, and the choice of form often depends on the context. For example, in large-scale quadratic programs, it can be algorithmically advantageous to assume that the constraints are posed in “standard form”, in which all general constraints are equalities, and the only inequalities are simple upper and lower bounds on the variables.

To simplify the notation in this paper, we consider only general lower-bound inequality constraints; however, the methods to be described can be generalized to treat all forms of linear constraints. The quadratic program to be solved is thus

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & \varphi(x) = c^Tx + \tfrac{1}{2}x^THx \\ \text{subject to} & \mathcal{A}x \geq \beta, \end{array} \tag{1.1}$$

where the *Hessian matrix* H is symmetric, and $\mathcal{A}$ is an $m_L \times n$ matrix. Any point x satisfying $\mathcal{A}x \geq \beta$ is said to be *feasible*. The gradient of φ is the linear function $g(x) = c + Hx$. When H is known to be positive definite, (1.1) is called a *convex* QP; when H may be any symmetric matrix, (1.1) is said to be a *general* QP.

This paper has two main purposes: first, to present an overview of the theoretical properties of a class of active-set methods for general quadratic programs; and second,

*This paper originally appeared in *SIAM Review*, Volume 33, Number 1, 1991, pages 1–36. The bibliography and author information have been updated.

†Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu). Research supported by the National Science Foundation grants DMS-9973276, and CCF-0082100.

‡Department of Management Science and Engineering, Stanford University, Stanford, CA 94305-4026 (walter@stanford.edu, saunders@stanford.edu). The research of these authors was supported by National Science Foundation grants DMI-9500668, CCR-9988205, and CCR-0306662, and Office of Naval Research grants N00014-96-1-0274 and N00014-02-1-0076.

§Department of Computer Science, Courant Institute, New York University, New York, NY 10012 (mhw@cs.nyu.edu).

to specify the equations and recurrence relations satisfied by the search direction and Lagrange multipliers. At each iteration of an active-set method, a certain subset of the constraints (the *working set*) is of central importance. The definitive feature of the class of methods considered (which we call *inertia-controlling*) is that the strategy for choosing the working set ensures that the reduced Hessian with respect to the working set (see Section 2.3) never has more than one nonpositive eigenvalue. In contrast, certain methods for general quadratic programming allow any number of nonpositive eigenvalues in the reduced Hessian—for example, the methods of Murray [33] and Bunch and Kaufman [3].

Our major focus will be on issues arising in *general* quadratic programming, and we shall not examine details of the many methods proposed for the convex case. Because the reduced Hessian is always positive definite for a convex QP, there is no need to impose an inertia-controlling strategy or to invoke complicated results involving singularity and indefiniteness. Nonetheless, many of the recurrence relations developed in Section 5 may be applied directly in methods for convex QP. A recent review of active-set quadratic programming methods is given in [14].

To our knowledge, Fletcher's method [12] was the first inertia-controlling quadratic programming method, and is derived using the partitioned inverse of the Karush-Kuhn-Tucker matrix (see Sections 2.3 and 5.1). His original paper and subsequent book [13] discuss many of the properties to be considered here. The methods of Gill and Murray [18] and of QPSOL [22] are inertia-controlling methods in which the search direction is obtained from the Cholesky factorization of the reduced Hessian matrix. Gould [29] proposes an inertia-controlling method for sparse problems, based on updating certain *LU* factorizations. Finally, the Schur-complement QP methods of Gill et al. [21, 25] are designed mainly for sparse problems, particularly those associated with applying Newton-based sequential quadratic programming (SQP) methods to large nonlinearly constrained problems.

If suitable initial conditions apply and H is positive definite, an identical sequence of iterates will be generated by inertia-controlling methods and by a wide class of theoretically equivalent methods for convex QP (see, e.g., Cottle and Djang [9]). Similarly, the methods of Murray [33] and Bunch and Kaufman [3] will generate the same iterates as inertia-controlling methods when solving a *general* QP if certain conditions hold. Despite these theoretical similarities, inertia-controlling methods are important in their own right because of the useful *algorithmic* properties that follow when the reduced Hessian has at most one nonpositive eigenvalue. In particular, the system of equations that defines the search direction has the same structure regardless of the eigenvalues of the reduced Hessian; this consistency allows certain factorizations to be recurred efficiently (see Section 6).

We shall consider only *primal-feasible* ("primal") QP methods, which require an initial feasible point x_0, and thereafter generate a sequence $\{x_k\}$ of feasible approximations to the solution of (1.1). If the feasible region of (1.1) is non-empty, a feasible point to initiate the QP iterations can always be found by solving a linear programming problem in which the (piecewise linear) sum of infeasibilities is minimized. (This procedure constitutes the *feasibility phase*, and will not be discussed here; see, e.g., Dantzig [10].) Despite our restriction, it should be noted that an inertia-controlling strategy of imposing an explicit limit on the number of nonpositive eigenvalues of the reduced Hessian can be applied in QP methods that do not require feasibility at every iteration (e.g., in the method of Hoyle [31]). We briefly discuss *dual* QP methods at the end of Section 2.1.

Before proceeding, we emphasize that any discussion of QP methods should distinguish between *theoretical* and *computational* properties. Even if methods are based on mathematically identical definitions of the iterates, their performance in practice depends on the efficiency, storage requirements and stability of the associated *numerical* procedures. Various mathematical equivalences among QP methods are discussed in Cottle and Djang [9] and Best [2]. In the present paper, Sections 2–4 are concerned primarily with theory, and Sections 5–6 treat computational matters.

The remainder of this paper is organized as follows. Section 2 summarizes the algorithmic structure of active-set QP methods. In Section 3, we present theoretical background needed to prove the crucial features of inertia-controlling methods, which are collected in Section 4. Section 5 contains recurrence relations of importance in implementation, without the restriction imposed in earlier treatments that the initial point must be a minimizer. Two particular methods are described in Section 6, including a new method in Section 6.1. Some brief conclusions and directions for future research are mentioned in Section 7.

2. Inertia-Controlling Active-Set Methods.

2.1. Optimality conditions. The point x is a local optimal solution of (1.1) if there exists a neighborhood of x such that $\varphi(x) \le \varphi(\bar{x})$ for *every feasible point* $\bar{x}$ in the neighborhood. To ensure that x satisfies this definition, it is convenient to verify certain *optimality conditions* that involve the relationship between φ and the constraints.

The vector p is called a *direction of decrease* for φ at x if there exists $\tau_\varphi > 0$ such that $\varphi(x + \alpha p) < \varphi(x)$ for all $0 < \alpha < \tau_\varphi$. Every suitably small positive step along a direction of decrease thus produces a strict reduction in φ. The nonzero vector p is said to be a *feasible direction* for the constraints of (1.1) at x if there exists $\tau_A > 0$ such that $x + \alpha p$ is feasible for all $0 < \alpha \le \tau_A$, i.e., if feasibility is retained for every suitably small positive step along p. If a feasible direction of decrease exists at x, every neighborhood of x must contain feasible points with a strictly lower value of φ, and consequently x cannot be an optimal solution of (1.1).

The optimality conditions for (1.1) involve the subset of constraints *active* or *binding* (satisfied exactly) at a possible solution x. (If a constraint is inactive at x, it remains satisfied in every sufficiently small neighborhood of x.) Let $\mathcal{I}_B$ ("B" for "binding") be the set of indices of the constraints active at the point x, and let A_B denote the matrix whose rows are the normals of the active constraints. (Both $\mathcal{I}_B$ and A_B depend on x, but this dependence is usually omitted to simplify notation.)

The following conditions are *necessary* for the feasible point x to be a solution of (1.1):

$$g(x) = A_B^T \mu_B \quad \text{for some } \mu_B; \tag{2.1a}$$

$$\mu_B \ge 0; \tag{2.1b}$$

$$v^T H v \ge 0 \quad \text{for all vectors } v \text{ such that } A_B v = 0. \tag{2.1c}$$

The necessity of these conditions is usually proved by contradiction: if all three are not satisfied at an alleged optimal point x, a feasible direction of decrease must exist, and x cannot be optimal.

The vector μ_B in (2.1a) is called the vector of *Lagrange multipliers* for the active constraints, and is unique only if the active constraints are linearly independent. Let Z_B denote a basis for the null space of A_B, i.e., every vector v satisfying $A_B v = 0$ can be written as a linear combination of the columns of Z_B. (Except in the trivial case, Z_B is not unique.) The vector $Z_B^T g(x)$ and the matrix $Z_B^T H Z_B$ are called the *reduced gradient* and *reduced Hessian* of φ (with respect to A_B). Condition (2.1a) is equivalent to the requirement that $Z_B^T g(x) = 0$, and (2.1c) demands that $Z_B^T H Z_B$ be positive semidefinite. Satisfaction of (2.1a) and (2.1c) is independent of the choice of Z_B.

Various *sufficient* optimality conditions for (1.1) can be stated, but the following are most useful for our purposes. The feasible point x is a solution of (1.1) if there exists a *subset* $\mathcal{I}_P$ of $\mathcal{I}_B$ ("P" for *positive* multipliers and *positive* definite), with corresponding matrix A_P of constraint normals, such that

(2.2a) $$g(x) = A_P^T \mu_P;$$

(2.2b) $$\mu_P > 0;$$

(2.2c) $$v^T H v > 0 \quad \text{for all nonzero vectors } v \text{ such that } A_P v = 0.$$

Condition (2.2b) states that all Lagrange multipliers associated with A_P are positive, and (2.2c) is equivalent to positive-definiteness of the reduced Hessian $Z_P^T H Z_P$, where Z_P denotes a basis for the null space of A_P. When the sufficient conditions hold, x is not only optimal, but is also *locally unique*, i.e., $\varphi(x) < \varphi(\bar{x})$ for all feasible $\bar{x}$ in a neighborhood of x ($\bar{x} \neq x$).

The gap between (2.1) and (2.2) arises from the possibility of one or more zero Lagrange multipliers and/or a reduced Hessian that is positive semidefinite and singular. When the necessary conditions are satisfied at some point x but the sufficient conditions are not, a feasible direction of decrease may or may not exist, so that x is not necessarily a local solution of (1.1). (For example, consider minimizing $x_1 x_2$ subject to $x_1 \geq 0$ and $x_2 \geq 0$. The origin is a local solution of this problem, but is not a solution when minimizing $-x_1 x_2$ subject to the same constraints.) Verification of optimality in such instances requires further information, and is in general an NP-hard problem (see Murty and Kabadi [34], Pardalos and Schnitger [36]). An alternative (equivalent) issue arises in the *copositivity problem* of quadratic programming (see, e.g., Contesse [7], Majthay [32]). A computational procedure for verifying optimality in the context of inertia-controlling methods is given in [17].

In active-set *dual* QP methods, a sequence of *infeasible* iterates $\{x_k\}$ is generated whose associated multipliers $\{\mu_k\}$ are *dual feasible*, i.e., satisfy the optimality conditions (2.1a–2.1b). We have not treated dual methods because all such methods known to us are restricted to QP problems in which H is positive semidefinite, and are not inertia-controlling in the sense defined here. For example, the method of Goldfarb and Idnani [26] has been extended by Stoer [38] to quadratic programs in which the objective function represents a (possibly rank-deficient) least-squares problem (i.e., $H = C^T C$, and the vector c from the objective lies in the range of C). The first step of Stoer's method involves finding the minimum-length solution of the unconstrained least-squares problem, and hence the reduced Hessian may have any number of zero eigenvalues. The main results of this paper would apply to dual methods that include an inertia-controlling strategy.

2.2. Definition of an iteration. In common with many optimization algorithms, inertia-controlling QP methods generate a sequence $\{x_k\}$ of approximations to the solution of (1.1) that satisfy

$$x_{k+1} = x_k + \alpha_k p_k.$$

(Any reference hereinafter to "the algorithm" means a generic inertia-controlling method.) The *search direction* p_k is an n-vector that is either zero or a direction of decrease, and the scalar *steplength* α_k is nonnegative. We usually consider a typical iteration (the k-th), and use unsubscripted symbols to denote quantities associated with iteration k when the meaning is clear.

Let g denote $g(x)$, the gradient of φ at the current iterate. The following (standard) terminology is useful in characterizing the relationship between p and φ:

$$p \text{ is a } \begin{cases} \text{descent direction} & \text{if } g^Tp < 0; \\ \text{direction of positive curvature} & \text{if } p^THp > 0; \\ \text{direction of negative curvature} & \text{if } p^THp < 0; \\ \text{direction of zero curvature} & \text{if } p^THp = 0. \end{cases}$$

Because φ is quadratic,

$$\varphi(x + \alpha p) = \varphi(x) + \alpha g^Tp + \tfrac{1}{2}\alpha^2 p^THp, \tag{2.3}$$

which shows that every direction of decrease p must be either a descent direction, or a direction of negative curvature with $g^Tp = 0$. If $g^Tp < 0$ and $p^THp > 0$, we see from (2.3) that $\varphi(x + \alpha p) < \varphi(x)$ for all $0 < \alpha < \tau$, where $\tau = -2g^Tp/p^THp$. If $g^Tp < 0$ and $p^THp \le 0$, or if $g^Tp = 0$ and $p^THp < 0$, (2.3) shows that φ is monotonically decreasing along p, i.e., $\varphi(x + \alpha p) < \varphi(x)$ for *all* $\alpha > 0$.

2.3. The role of the working set. At iteration k, p_k is defined in terms of a subset of the constraints, designated as the *working set*. The matrix whose rows are normals of constraints in the working set at x_k will be called A_k. The "new" working set A_{k+1} is always obtained by modifying the "old" working set, and the prescription for altering the working set is known for historical reasons as the *active-set strategy*.

A crucial element in any formal description of an active-set QP method is the precise definition of an "iteration" (in effect, the decision as to when the iteration counter k is incremented). As we shall see, the working set can change when x does not, and vice versa. To avoid ambiguity, we adopt the convention that the iteration counter changes when *either* the point x or the working set changes. An inertia-controlling method thus generates a sequence of distinct *pairs* $\{x_k, A_k\}$, where $x_{k+1} \ne x_k$ *or* $A_{k+1} \ne A_k$. (In some circumstances, both iterate and working set change, i.e. $x_{k+1} \ne x_k$ and $A_{k+1} \ne A_k$.) It may seem unappealing to label the same point or matrix with different subscripts, but other numbering conventions can lead to indeterminacy in the specification of either x_k or A_k for a given value of k.

Although it is sometimes useful to think of the working set as a prediction of the set of constraints active at the solution of (1.1), we stress that this interpretation may be misleading. The working set is defined *by the algorithm*, not simply by the active constraints. In particular, the working set may not contain all the active constraints at any iterate, including the solution.

Dropping the subscript k, we let m denote the number of rows of A, $\mathcal{I}$ the set of indices of constraints in the working set, and b the vector of corresponding components of β. We refer to both the index set $\mathcal{I}$ and the matrix A as the "working set".

Let Z denote a matrix whose columns form a basis for the null space of A; the reduced gradient and reduced Hessian of φ with respect to A are then $Z^Tg(x)$ and Z^THZ. We sometimes denote the reduced Hessian by H_Z. A nonzero vector v such that $Av = 0$ is called a *null-space direction*, and can be written as a linear combination of the columns of Z.

In inertia-controlling methods, the working set is constructed to have three important characteristics:

WS1. Constraints in the working set are active at x;
WS2. The rows of the working set are linearly independent;
WS3. The working set at x_0 is chosen so that the initial reduced Hessian is positive definite.

Although each of these properties has an essential role in proving that an inertia-controlling algorithm is well defined (see Sections 3 and 4), some of them also apply to other active-set methods.

We emphasize that it may not be possible to enforce the crucial property WS3 at an arbitrary starting point x_0 if the working set is selected only from the "original" constraints—for example, suppose that H is indefinite and no constraints are active at x_0. Inertia-controlling methods must therefore include the ability to add certain "temporary" constraints to the initial working set in order to ensure that property WS3 holds. Such constraints are an algorithmic device, and do not alter the solution (see Section 4.4).

This paper will consider only active-set primal-feasible methods that require property WS1 to apply at the next iterate $x + \alpha p$ with the *same* working set used to define any nonzero p. This additional condition implies that the search direction must be *a null-space direction*, so that $Ap = 0$, and the methods of this paper are sometimes described as *null-space methods*.

A *stationary point* of the original QP (1.1) with respect to a particular working set A is any feasible point x for which $Ax = b$ and the gradient of the objective function is a linear combination of the columns of A^T, i.e.,

$$g = c + Hx = A^T \mu, \tag{2.4}$$

where $g = g(x)$. Since A has full row rank, μ is unique. For any stationary point, let μ_s ("s" for "smallest") denote the minimum component of μ, i.e., $\mu_s = \min \mu_i$. An equivalent statement of (2.4) is that the reduced gradient is zero at any stationary point. The *Karush-Kuhn-Tucker* (KKT) matrix K corresponding to A is defined by

$$K \equiv \begin{pmatrix} H & A^T \\ A & \end{pmatrix}. \tag{2.5}$$

When the reduced Hessian is nonsingular, K is nonsingular (see Corollary 3.2).

A stationary point at which the reduced Hessian is positive definite is called a *minimizer* (with respect to A), and is the unique solution of a QP in which constraints in the working set appear as *equalities*:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^T x + \tfrac{1}{2} x^T H x \\ \text{subject to} & Ax = b. \end{array} \tag{2.6}$$

The Lagrange multiplier vector for the equality constraints of (2.6) is the vector μ of (2.4). When the reduced Hessian is positive definite, the solution of (2.6) is $x - q$, where q solves the *KKT system*

$$K \begin{pmatrix} q \\ \mu \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{2.7}$$

and μ is the associated Lagrange multiplier vector. If x is a stationary point, $q = 0$.

Given an iterate x and a working set A, an inertia-controlling method must be able to

- determine whether x is a stationary point with respect to A;
- calculate the (unique) Lagrange multiplier vector μ at stationary points (see (2.4));
- determine whether the reduced Hessian is positive definite, positive semidefinite and singular, or indefinite.

In the present theoretical context, we simply assume this ability; Sections 5–6 discuss techniques for computing the required quantities.

To motivate active-set QP methods, it is enlightening to think in terms of desirable properties of the search direction. For example, since p is always a null-space direction (i.e., $Ap = 0$), any step along p stays "on" constraints in the working set. Furthermore, it seems "natural" to choose p as a direction of decrease for φ because problem (1.1) involves minimizing φ. We therefore seek to obtain a null-space direction of decrease, which can be computed using the current working set in the following two situations:

(i) when x is not a stationary point;

(ii) when x is a stationary point and the reduced Hessian is indefinite.

If neither (i) nor (ii) applies, the algorithm terminates or changes the working set (see Section 2.4).

When (i) holds, the nature of p depends on the reduced Hessian. (The specific equations satisfied by p are given in Section 4.1; only its general properties are summarized here.) If the reduced Hessian is positive definite, p is taken as $-q$, the necessarily nonzero step to the solution of the associated equality-constrained subproblem (see (2.6) and (2.7)). This vector is a descent direction of positive curvature, and has the property that $\alpha = 1$ is the step to the smallest value of φ along p. When the reduced Hessian is positive semidefinite and singular, p is chosen as a descent direction of zero curvature. When the reduced Hessian is indefinite, p is taken as a descent direction of negative curvature.

When (ii) holds, i.e., when x is a stationary point with an indefinite reduced Hessian, p is taken as a direction of negative curvature.

2.4. Deleting constraints from the working set. When x is a stationary point at which the reduced Hessian is positive semidefinite, it is impossible to reduce φ by moving along a null-space direction. Depending on the sign of the smallest Lagrange multiplier and the nature of the reduced Hessian, the algorithm must either terminate or change the working set by deleting one or more constraints.

Let x be any stationary point (so that $g = A^T\mu$), and suppose that $\mu_s < 0$ for constraint s in the working set. Let e_s be the s-th coordinate vector. Given a vector p satisfying

$$Ap = \gamma e_s \quad \text{with} \quad \gamma > 0,$$

a positive step along p moves "off" (strictly feasible to) constraint s, but remains "on" the other constraints in A. (The full rank of the working set guarantees that the equations $Ap = v$ are compatible for any vector v.) It follows that

$$g^Tp = \mu^T Ap = \gamma\mu^T e_s = \gamma\mu_s < 0,$$

so that p is a descent direction. A negative multiplier for constraint s thus suggests that a null-space descent direction can be found by *deleting* constraint s from the working set. However, our freedom to delete constraints is limited by the inertia-controlling strategy. To ensure that the reduced Hessian has no more than one non-positive eigenvalue, *a constraint can be deleted only at a minimizer.* (Section 3.3

provides theoretical validation of this policy.) Following deletion of a constraint at a minimizer, the reduced Hessian (with respect to the smaller working set) can have *at most one* nonpositive eigenvalue (see Lemma 3.4).

When x is a minimizer, the action of the algorithm depends on the nature of μ_s, the smallest multiplier. There are three cases to consider.

1. $\mu_s > 0$**.** The sufficient conditions (2.2) for optimality apply with $\mathcal{I}_P = \mathcal{I}$, and the algorithm terminates.

2. $\mu_s < 0$**.** Constraint s is deleted from the working set; x cannot then be a stationary point with respect to the "new" working set.

3. $\mu_s = 0$**.** The current iterate x may or may not be a solution of the original problem (1.1); see the discussion in Section 2.1. We mention two possible strategies for treating the "zero multiplier" case.

Strategy Z1. If $\mu_s = 0$ at a minimizer, an inertia-controlling algorithm may choose to *terminate*, since the necessary conditions (2.1) for optimality are satisfied at x, with zero multipliers corresponding to any idle constraints. (A constraint that is active at x but is not in the working set is called *idle*.) This strategy has several virtues, including simplicity. As noted by Fletcher [12], x is the unique solution of a QP whose objective function is a *small perturbation* of φ. If the full Hessian H happens to be positive semidefinite, termination is appropriate because x is necessarily a solution of the original problem.

Strategy Z2. Alternatively, an inertia-controlling algorithm can *delete* a constraint when $\mu_s = 0$; this strategy is used in QPSOL ([22]). The uniqueness of μ implies not only that x stays a stationary point after removal of the chosen constraint, but also that the multipliers corresponding to the remaining constraints are unaltered. The algorithm may therefore continue to delete constraints with zero multipliers until (i) a working set is found for which $\mu_s > 0$, or (ii) the reduced Hessian ceases to be positive definite. If the reduced Hessian is positive definite after all constraints with zero multipliers have been deleted, x satisfies the sufficient optimality conditions (2.2) and the algorithm may terminate with an assured solution. Once the reduced Hessian has ceased to be positive definite, the inertia-controlling strategy dictates that no further constraints may be deleted.

Strategy Z2 does *not* ensure that a valid solution will be found; as indicated in Section 2.1, the problem of verifying optimality under these circumstances is NP-hard. However, it enlarges the set of problems for which the algorithm can successfully move away from a non-optimal point—for example, let x be the origin when minimizing $x_1x_2 + \frac{1}{2}x_2^2$ subject to $x_1 \geq 0$ and $x_1 + x_2 \geq 0$.

With finite precision, it is impossible to devise an infallible *numerical* test for a "zero" multiplier. But since a decision as to what constitutes "numerical zero" must be made in any case to distinguish between "negative" and "nonnegative" multipliers, the same criterion can be applied to designate a "negligible" multiplier.

Following a nonzero step and a sequence of constraint additions, the next iterate of an inertia-controlling method can be a stationary point only if the reduced Hessian is *positive definite* (see Lemma 4.5). The only situation in which an iterate can be a stationary point with a reduced Hessian that is positive semidefinite and singular occurs when the present working set was obtained by deleting a constraint with a zero multiplier from the previous working set; after such a deletion, the smallest multiplier must be nonnegative (otherwise, it would have been deleted previously), and the algorithm terminates.

The pseudo-code in Figure 1 summarizes a constraint deletion procedure for the k-th iteration, where we give the iteration count explicitly for clarity. Figure 1 treats the zero-multiplier case with strategy Z2; an obvious change in the test on μ_s in the third line would give the pseudo-code corresponding to strategy Z1.

The logical variables *positive_definite*, *positive_semidefinite* and *singular* are assumed to be computed before starting the iteration; the logical variable *complete* is used to terminate the overall algorithm (see Figure 3). The details of the boxed computation (deleting a constraint from the working set) depend on the particular inertia-controlling algorithm (see Section 5.1). When a constraint is deleted, the working set is altered while x remains unchanged.

```
if stationary_point and positive_semidefinite then
    μ_s ← smallest component of μ_k;
    if singular or μ_s > 0 then
        complete ← true
    else
        | delete constraint s from the working set; |
        I_{k+1} ← I_k − {s}; x_{k+1} ← x_k; k ← k + 1;
    end if
end if
```

Figure 1. Pseudo-code for constraint deletion.

2.5. Adding constraints to the working set. Constraints are deleted from the working set until the algorithm terminates or a nonzero p can be defined. Since any nonzero p is always a direction of decrease, the goal of minimizing φ suggests that the steplength α should be taken as the step along p that produces the largest decrease in φ. Because p is a null-space direction, $x + \alpha p$ automatically remains feasible with respect to constraints in the working set. However, α may need to be restricted to retain feasibility with respect to constraints *not* in the working set, which are added to the working set.

Let i be the index of a constraint not in the working set. The constraint will not be violated at $x + \alpha p$ for any positive step α if $a_i^T p \geq 0$. If $a_i^T p < 0$, however, the constraint will become active at a certain nonnegative step. For every $i \notin \mathcal{I}$, α_i is defined as

$$\alpha_i = \begin{cases} (\beta_i - a_i^T x)/a_i^T p & \text{if } a_i^T p < 0; \\ +\infty & \text{otherwise.} \end{cases} \tag{2.8}$$

The *maximum feasible step* α_F (often called the *step to the nearest constraint*) is defined as $\alpha_F \equiv \min \alpha_i$. The value of α_F is zero if and only if $a_i^T p < 0$ for at least one idle constraint i. If α_F is infinite, the constraints do not restrict positive steps along p.

In order to retain feasibility, α must satisfy $\alpha \leq \alpha_F$. If the reduced Hessian is positive definite, the step of unity along p has special significance, since p in this case is taken as $-q$ of (2.7), and φ achieves its minimum value along p at $\alpha = 1$ (see (2.6)).

When the reduced Hessian is either indefinite or positive semidefinite and singular, φ is monotonically decreasing along p (see Section 2.2). Hence, the nonnegative step α along p that produces the maximum reduction in φ and retains feasibility is

$$\alpha = \begin{cases} \min(1, \alpha_F) & \text{if } Z^THZ \text{ is positive definite;} \\ \alpha_F & \text{otherwise.} \end{cases}$$

In order for the algorithm to proceed, α must be finite. If $\alpha = \infty$, φ is unbounded below in the feasible region, (1.1) has an infinite solution, and the algorithm terminates.

Let r denote the index of a constraint for which $\alpha_F = \alpha_r$. The algorithm requires a single value of r, so that some rule is necessary in case of ties—for example, r may be chosen to improve the estimated condition of the working set. (Several topics related to this choice are discussed in Gill et al. [24].) When $\alpha = \alpha_F$, the constraint with index r becomes active at the new iterate. In the inertia-controlling methods to be considered, a_r is *added* to the working set at this stage of the iteration, with one exception: a constraint is *not* added when the reduced Hessian is positive definite and $\alpha_F = 1$. In this case, $x + p$ is automatically a minimizer with respect to the current working set, which means that at least one constraint will be deleted at the beginning of the next iteration (see Section 2.4).

Assuming the availability of a suitable direction of decrease p, the pseudo-code in Figure 2 summarizes the constraint addition procedure during iteration k. As in Figure 1, details of the boxed computation (adding a constraint to the working set) depend on the particular inertia-controlling algorithm (see, e.g., Sections 6.1 and 6.2). Even following a constraint addition, x_{k+1} may be the same as x_k if $\alpha_k = 0$.

```
α_F ← maximum feasible step along p_k (to constraint r);
hit_constraint ← not positive_definite or α_F < 1;
if hit_constraint then α_k ← α_F else α_k ← 1 end if;
if α = ∞ then stop
else
    x_{k+1} ← x_k + α_k p_k;
    if hit_constraint then
        [add constraint r to the working set;]
        I_{k+1} ← I_k ∪ {r}
    else
        I_{k+1} ← I_k
    end if
    k ← k + 1;
end if
```

Figure 2. Pseudo-code for constraint addition.

The following lemma shows that all working sets have full rank in methods that satisfy the rules given above for choosing the initial working set and adding constraints to the working set.

LEMMA 2.1. *Assume that the initial working set has full row rank. For an active-set QP algorithm of the form described, any constraint added to the working set must be linearly independent of the constraints in the working set.*

Proof. A constraint a_r can be added to the working set A only if $a_r^T p < 0$ (see (2.8)). If a_r were linearly dependent on the working set, we could express a_r as $a_r = A^T r$ for some nonzero vector r. However, p is a null-space direction, and the relation $Ap = 0$ would then imply that $a_r^T p = r^T Ap = 0$, a contradiction. ∎

Putting together the deletion and addition strategies, Figure 3 summarizes the general structure of an inertia-controlling QP method. The logical variable *complete* indicates whether the method has terminated.

```
complete ← false; k ← 0;
repeat until complete
    execute constraint deletion procedure (Figure 1);
    if not complete then
        ┌─────────────┐
        │ compute p_k; │
        └─────────────┘
        if p_k ≠ 0 then
            execute constraint addition procedure (Figure 2)
        end if
    end if
end repeat
```

Figure 3. Structure of an inertia-controlling method.

3. Theoretical Background. This section summarizes theoretical results used in proving that inertia-controlling methods are well defined.

3.1. The Schur complement. Let T be the partitioned symmetric matrix

$$T = \begin{pmatrix} M & W^T \\ W & G \end{pmatrix}, \tag{3.1}$$

where M is nonsingular, and M and G are symmetric. The *Schur complement* of M in T, denoted by T/M, is defined as

$$T/M \equiv G - WM^{-1}W^T. \tag{3.2}$$

We sometimes refer simply to "the" Schur complement when the relevant matrices are evident.

An important application of the Schur complement is in solving $Ty = d$ when T has the form (3.1) and is nonsingular. Let the right-hand side d and the unknown y be partitioned to conform with (3.1):

$$d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \qquad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

Then y may be obtained by solving (in order)

$$Mw = d_1 \tag{3.3a}$$

$$(T/M)y_2 = d_2 - Ww \tag{3.3b}$$

$$My_1 = d_1 - W^T y_2. \tag{3.3c}$$

A Schur complement analogous to (3.2) can be defined for a nonsymmetric matrix T. When M is singular, the *generalized Schur complement* is obtained by substituting the generalized inverse of M for M^{-1} in (3.2), and by appropriate adjustment of (3.3). The "classical" Schur complement (3.2) and its properties are discussed in detail by Cottle [8]. For further details on the generalized Schur complement, see Carlson, Haynsworth and Markham [6] and Ando [1]. Carlson [5] gives an interesting survey of results on both classical and generalized Schur complements, along with an extensive bibliography.

Let S be any symmetric matrix. We denote by $i_p(S)$, $i_n(S)$ and $i_z(S)$ respectively the (nonnegative) numbers of positive, negative and zero eigenvalues of S. The *inertia* of S—denoted by $\mathrm{In}\,(S)$—is the associated integer triple (i_p, i_n, i_z). For any suitably dimensioned nonsingular matrix N, *Sylvester's law of inertia* states that

$$\mathrm{In}\,(S) = \mathrm{In}\,(N^TSN). \tag{3.4}$$

The inertias of M and T from (3.1) and the Schur complement T/M of (3.2) satisfy the following important equation:

$$\mathrm{In}\,(T) = \mathrm{In}\,(M) + \mathrm{In}\,(T/M) \tag{3.5}$$

(see Haynsworth [30]).

3.2. The KKT matrix and the reduced Hessian. The eigenvalue structure of the reduced Hessian determines the logic of an inertia-controlling method, and the KKT matrix of (2.5) plays a central role in defining the search direction. The following theorem gives an important relationship between the KKT matrix and the reduced Hessian Z^THZ.

THEOREM 3.1. *Let H be an $n \times n$ symmetric matrix, A an $m \times n$ matrix of full row rank, K the KKT matrix of (2.5), and Z a null-space basis for A. Then*

$$\mathrm{In}\,(K) = \mathrm{In}\begin{pmatrix} H & A^T \\ A & \end{pmatrix} = \mathrm{In}\,(Z^THZ) + (m, m, 0).$$

Proof. See Gould [27]. Since every basis for the null space may be written as ZS for some nonsingular matrix S, Sylvester's law of inertia (3.4) implies that the inertia of the reduced Hessian is independent of the particular choice of Z. We emphasize that the full rank of A is essential in this result. □

COROLLARY 3.2. *The KKT matrix K is nonsingular if and only if the reduced Hessian Z^THZ is nonsingular.* □

3.3. Changes in the working set. The nature of the KKT matrix leads to several results concerning the eigenvalue structure of the reduced Hessian following a change in the working set.

LEMMA 3.3. *Let M and M_+ denote symmetric matrices of the following form:*

$$M = \begin{pmatrix} H & B^T \\ B & \end{pmatrix} \quad \text{and} \quad M_+ = \begin{pmatrix} H & B_+^T \\ B_+ & \end{pmatrix},$$

where B_+ is B with one additional row. (The subscript "+" is intended to emphasize which matrix has the extra row.) Then exactly one of the following cases holds:

(a) $i_p(M_+) = i_p(M) + 1,\ i_n(M_+) = i_n(M)$ *and* $i_z(M_+) = i_z(M)$*;*
(b) $i_p(M_+) = i_p(M) + 1,\ i_n(M_+) = i_n(M) + 1$ *and* $i_z(M_+) = i_z(M) - 1$*;*
(c) $i_p(M_+) = i_p(M),\ i_n(M_+) = i_n(M) + 1$ *and* $i_z(M_+) = i_z(M)$*;*
(d) $i_p(M_+) = i_p(M),\ i_n(M_+) = i_n(M)$ *and* $i_z(M_+) = i_z(M) + 1$*.*

Proof. It is sufficient to prove the result for the case when

$$B_+ = \begin{pmatrix} B \\ b^T \end{pmatrix}, \tag{3.6}$$

where b^T is a suitably dimensioned row vector. If the additional row of B_+ occurs in any position other than the last, there exists a permutation Π (representing a row interchange) such that ΠB_+ has the form (3.6). Let

$$P = \begin{pmatrix} I & \\ & \Pi \end{pmatrix}, \quad \text{which gives} \quad PM_+P^T = \begin{pmatrix} H & B^T & b \\ B & & \\ b^T & & \end{pmatrix}. \tag{3.7}$$

Because P is orthogonal, PM_+P^T is a similarity transform of M_+, and has the same eigenvalues (see Wilkinson [39], page 7). Thus the lemma applies equally to M_+ and PM_+P^T.

When B_+ has the form (3.6), standard theory on the interlacing properties of the eigenvalues of bordered symmetric matrices states that

$$\lambda_1^+ \ge \lambda_1 \ge \lambda_2^+ \ge \cdots \ge \lambda_\ell \ge \lambda_{\ell+1}^+,$$

where ℓ is the dimension of M, and $\{\lambda_i\}$ and $\{\lambda_i^+\}$ are the eigenvalues of M and M_+ respectively, in decreasing order (see, e.g., Wilkinson [39], pages 96–97). The desired results follow by analyzing the consequences of these inequalities. □

By combining the general interlacing result of Lemma 3.3 with the specific properties of the KKT matrix from Theorem 3.1, we derive the following lemma, which applies to either adding or deleting a single constraint from the working set.

LEMMA 3.4. *Let A be an $m \times n$ matrix of full row rank, and let A_+ denote A with one additional linearly independent row (so that A_+ also has full row rank). The matrices Z and Z_+ denote null-space bases for A and A_+, and H_Z and H_{Z_+} denote the associated reduced Hessian matrices Z^THZ and $Z_+^THZ_+$. (Note that the dimension of H_{Z_+} is one less than the dimension of H_Z.) Define K and K_+ as*

$$K = \begin{pmatrix} H & A^T \\ A & \end{pmatrix} \quad \text{and} \quad K_+ = \begin{pmatrix} H & A_+^T \\ A_+ & \end{pmatrix},$$

where H is an $n \times n$ symmetric matrix. Then exactly one of the following cases holds:
(a) $i_p(H_{Z_+}) = i_p(H_Z) - 1$, $i_n(H_{Z_+}) = i_n(H_Z) - 1$ *and* $i_z(H_{Z_+}) = i_z(H_Z) + 1$;
(b) $i_p(H_{Z_+}) = i_p(H_Z) - 1$, $i_n(H_{Z_+}) = i_n(H_Z)$ *and* $i_z(H_{Z_+}) = i_z(H_Z)$;
(c) $i_p(H_{Z_+}) = i_p(H_Z)$, $i_n(H_{Z_+}) = i_n(H_Z) - 1$ *and* $i_z(H_{Z_+}) = i_z(H_Z)$;
(d) $i_p(H_{Z_+}) = i_p(H_Z)$, $i_n(H_{Z_+}) = i_n(H_Z)$ *and* $i_z(H_{Z_+}) = i_z(H_Z) - 1$.

Proof. Since A and A_+ have full row rank, Theorem 3.1 applies to both K and K_+, and gives $i_p(K) = i_p(H_z) + m$, $i_p(K_+) = i_p(H_{Z_+}) + m + 1$, $i_n(K) \ge m$ and $i_n(K_+) \ge m + 1$. Substituting from these relations into the four cases of Lemma 3.3, we obtain the desired results. □

When a constraint is *added* to the working set, A and A_+ correspond to the "old" and "new" working sets. Lemma 3.4 shows that adding a constraint to the working set either leaves unchanged the number of nonpositive eigenvalues of the reduced Hessian, or decreases the number of nonpositive eigenvalues by one. The following corollary lists the possible outcomes of adding a constraint to the working set.

COROLLARY 3.5. *Under the same assumptions as in Lemma 3.4:*

(a) *if Z^THZ is positive definite and a constraint is added to the working set, $Z_+^THZ_+$ must be positive definite;*
(b) *if Z^THZ is positive semidefinite and singular and a constraint is added to the working set, $Z_+^THZ_+$ may be positive definite or positive semidefinite and singular;*
(c) *if Z^THZ is indefinite and a constraint is added to the working set, $Z_+^THZ_+$ may be positive definite, positive semidefinite and singular, or indefinite.* □

For a constraint deletion, on the other hand, the roles of A and A_+ are reversed (K_+ is the "old" KKT matrix and K is the "new"). In this case, Lemma 3.4 shows that deleting a constraint from the working set can either leave unchanged the number of nonpositive eigenvalues of Z^THZ, or increase the number of nonpositive eigenvalues by one.

If constraints are deleted only when the reduced Hessian is positive definite, Lemma 3.4 validates the inertia-controlling strategy by ensuring that the reduced Hessian will never have more than one nonpositive eigenvalue following a deletion and any number of additions. *Accordingly, when the reduced Hessian matrix is hereafter described as "indefinite", it has a single negative eigenvalue, with all other eigenvalues positive; and when the reduced Hessian matrix is described as "singular", it has one zero eigenvalue, with all other eigenvalues positive.*

3.4. Relations involving the KKT matrix. We now prove several results that will be used in Section 4. It should be emphasized that the following lemma assumes nonsingularity of K_+, but *not* of K.

LEMMA 3.6. *Let A and A_+ be matrices with linearly independent rows, where A_+ is A with a row added in position s. Let K, Z, K_+ and Z_+ be defined as in Lemma 3.4. If K_+ is nonsingular, then*

$$\mathrm{In}\,(K) + (1,1,0) = \mathrm{In}\,(K_+) + \mathrm{In}\,(-\sigma),$$

where σ is the $(n+s)$-th diagonal element of K_+^{-1}, i.e., $\sigma = e_{n+s}^T K_+^{-1} e_{n+s}$.

Proof. Consider the matrix

$$K_{\mathrm{aug}} \equiv \begin{pmatrix} K_+ & e_{n+s} \\ e_{n+s}^T & \end{pmatrix},$$

where e_{n+s} is the $(n+s)$-th coordinate vector. Using definition (3.2) to obtain the Schur complement of K_+ in K_{aug}, we obtain

$$K_{\mathrm{aug}}/K_+ = -e_{n+s}^T K_+^{-1} e_{n+s} = -\sigma.$$

Since K_+ is nonsingular, relation (3.5) applies with K_{aug} and K_+ in the roles of T and M, and we have

$$\mathrm{In}\,(K_{\mathrm{aug}}) = \mathrm{In}\,(K_+) + \mathrm{In}\,(-\sigma). \tag{3.8}$$

Because of the special forms of K and K_+, it is possible to obtain an expression that relates the inertias of K and K_{aug}. By assumption, the new row of A_+ is row s (denoted by a_s^T). As in (3.7), a permutation matrix P can be symmetrically applied to K_{aug} so that a_s^T becomes the last row in the upper left square block of size $n+m+1$. Further permutations lead to the following symmetrically reordered version of K_{aug}:

$$\widetilde{K}_{\mathrm{aug}} \equiv \widetilde{P}^T K_{\mathrm{aug}} \widetilde{P} = \begin{pmatrix} 0 & 1 & a_s & 0 \\ 1 & 0 & 0 & 0 \\ a_s^T & 0 & H & A^T \\ 0 & 0 & A & \end{pmatrix},$$

where $\widetilde{P}$ is a permutation matrix. Since $\widetilde{K}_{\rm aug}$ is a symmetric permutation of $K_{\rm aug}$, the two matrices have the same eigenvalues, and hence

$$\text{In}\,(K_{\rm aug}) = \text{In}\,(\widetilde{K}_{\rm aug}). \tag{3.9}$$

The 2×2 matrix in the upper left-hand corner of $\widetilde{K}_{\rm aug}$ (denoted by E) is nonsingular, with eigenvalues ± 1, and satisfies

$$\text{In}\,(E) = (1, 1, 0) \quad \text{with} \quad E^{-1} = E = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Using (3.2), we next verify algebraically that the Schur complement of E in $\widetilde{K}_{\rm aug}$ is simply K:

$$\widetilde{K}_{\rm aug}/E = K - \begin{pmatrix} a_s^T & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} a_s & 0 \\ 0 & 0 \end{pmatrix} = K.$$

Since $\text{In}\,(K_{\rm aug}) = \text{In}\,(\widetilde{K}_{\rm aug})$ (from (3.9)) and $\text{In}\,(\widetilde{K}_{\rm aug}) = \text{In}\,(E) + \text{In}\,(\widetilde{K}_{\rm aug}/E)$ (from (3.5)), we obtain

$$\text{In}\,(K_{\rm aug}) = \text{In}\,(E) + \text{In}\,(\widetilde{K}_{\rm aug}/E) = (1, 1, 0) + \text{In}\,(K). \tag{3.10}$$

Combining (3.8) and (3.10) gives the desired result. ∎

The following two corollaries state connections between successive reduced Hessians and solutions of linear systems involving KKT matrices.

Corollary 3.7. *Let K and K_+ be defined as in Lemma 3.6. Consider the nonsingular linear system*

$$K_+ \begin{pmatrix} y \\ w \end{pmatrix} = e_{n+s}, \tag{3.11}$$

where y has n components. Let w_s denote the s-th component of w. (Since the solution of (3.11) is column $n+s$ of K_+^{-1}, $w_s = \sigma$ of Lemma 3.6.) Then:

(a) *if $Z^T HZ$ is positive definite and $Z_+^T HZ_+$ is positive definite, w_s must be negative;*

(b) *if $Z^T HZ$ is singular and $Z_+^T HZ_+$ is positive definite, w_s must be zero;*

(c) *if $Z^T HZ$ is indefinite and $Z_+^T HZ_+$ is positive definite, w_s must be positive.* ∎

Lemma 3.8. *Let K and K_+ be defined as in Lemma 3.6, with the further assumptions that $Z_+^T HZ_+$ is positive definite and $Z^T HZ$ is indefinite. Let z denote the first n components of the solution of*

$$K \begin{pmatrix} z \\ t \end{pmatrix} = \begin{pmatrix} H & A^T \\ A & \end{pmatrix} \begin{pmatrix} z \\ t \end{pmatrix} = \begin{pmatrix} a_s \\ 0 \end{pmatrix}, \tag{3.12}$$

where a_s^T is the additional row of A_+. Then $a_s^T z < 0$.

Proof. Because $Z^T HZ$ is indefinite, K is nonsingular (see Theorem 3.1). The vectors z and t of (3.12) are therefore unique, and satisfy

$$Hz + A^T t - a_s = 0, \quad Az = 0. \tag{3.13}$$

We now relate the solutions of (3.12) and (3.11). Because of the special structure of K_+, the unique solution of (3.11) satisfies

$$Hy + A^T w_A + a_s w_s = 0, \quad Ay = 0, \quad a_s^T y = 1, \tag{3.14}$$

where w_A denotes the subvector of w corresponding to A, and w_s is the component of w corresponding to a_s. Part (c) of Corollary 3.7 implies that $w_s > 0$. Comparing (3.14) and (3.13), we conclude that $y = w_s z$. Since $a_s^T y = 1$, this relation implies that $a_s^T z = -1/w_s < 0$, which is the desired result. □

4. Theoretical Properties of Inertia-Controlling Methods. In this section we give the equations used to define the search direction after the working set has been chosen (see Section 2.4), and then prove various properties of inertia-controlling methods. When the reduced Hessian is positive definite and x is not a minimizer, choosing p as $-q$ from the KKT system (2.7) means that $\alpha = 1$ (the step to the minimizer of φ along p) can be viewed as the "natural" step. In contrast, if the reduced Hessian is singular or indefinite, the search direction needs to be specified only to within a positive multiple. Since φ is monotonically decreasing along p when the reduced Hessian is not positive definite, the steplength α is determined not by φ, but by the *nearest constraint* (see Section 2.5). Hence, multiplying p by any positive number γ simply divides the steplength by γ, and produces the identical next iterate.

4.1. Definition of the search direction. The mathematical specification of the search direction depends on the eigenvalue structure of the reduced Hessian, and, in the indefinite case, on the nature of the current iteration.

Positive definite. If the reduced Hessian is positive definite, the search direction p is taken as $p = -q$, where q is part of the solution of the KKT system (2.7):

$$\begin{pmatrix} H & A^T \\ A & \end{pmatrix} \begin{pmatrix} q \\ \mu \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{4.1}$$

An equivalent definition of p, which will be relevant in Sections 6.1 and 6.2, involves the *null-space equations*:

$$p = Zp_Z, \quad \text{where} \quad Z^T HZp_Z = -Z^T g.$$

If x is a minimizer, $p = q = 0$.

Singular. If the reduced Hessian is singular and the algorithm does not terminate, we shall show later that x cannot be a stationary point (see Lemma 4.5). The search direction p is defined as $\beta\hat{p}$, where $\hat{p}$ is the unique nonzero direction satisfying

$$\begin{pmatrix} H & A^T \\ A & \end{pmatrix} \begin{pmatrix} \hat{p} \\ \nu \end{pmatrix} = 0 \tag{4.2}$$

and β is chosen to make p a descent direction. Equivalently, $\hat{p}$ is defined by

$$\hat{p} = Zp_Z, \quad \text{where} \quad Z^T HZp_Z = 0, \quad \|p_Z\| \neq 0,$$

where the vector p_Z is necessarily a multiple of the single eigenvector corresponding to the zero eigenvalue of $Z^T HZ$.

Indefinite. If the reduced Hessian is indefinite, it must be nonsingular, with exactly one negative eigenvalue. In this case, p is defined in two different ways.

First, if the current working set was obtained either by deleting a constraint with a *negative* multiplier from the immediately preceding working set, or by adding a constraint, then p is taken as q from the KKT system (2.7), i.e., p satisfies

$$\begin{pmatrix} H & A^T \\ A & \end{pmatrix} \begin{pmatrix} p \\ \mu \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{4.3}$$

Second, if the current working set is the result of deleting a constraint with a *zero* multiplier from the immediately preceding working set, let a_s denote the normal of the deleted constraint. The current point is still a stationary point with respect to A (see Section 2.4), and hence $g = A^T\mu$ for some vector μ. The search direction p is defined by

$$\begin{pmatrix} H & A^T & a_s \\ A & & \\ a_s^T & & \end{pmatrix} \begin{pmatrix} p \\ \nu \\ w_s \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 1 \end{pmatrix}, \tag{4.4}$$

which can also be written as

$$\begin{pmatrix} H & A^T & a_s \\ A & & \\ a_s^T & & \end{pmatrix} \begin{pmatrix} p \\ w \\ w_s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{4.5}$$

where $w = \nu - \mu$. The KKT matrix *including* a_s must have been nonsingular to allow a constraint deletion, so that the solution of either (4.4) or (4.5) is unique, and Corollary 3.7 implies that $w_s > 0$.

4.2. Intermediate iterations. Various properties of inertia-controlling methods have been proved by Fletcher and others (see, e.g., [12, 13, 18, 29]). In this section, we use the Schur-complement results of Section 3 to analyze certain sequences of iterates in an inertia-controlling method. The initial point x_0 is assumed to be feasible; the initial working set has full row rank and is chosen so that the reduced Hessian is positive definite (see Section 4.4).

The following terminology is intended to characterize the relationship between an iterate and a working set. Let x be an iterate of an inertia-controlling method and A a valid working set for x, i.e. the rows of A are *linearly independent* normals of constraints *active* at x. As usual, Z denotes a null-space basis for A. We say that

$$x \text{ is } \begin{cases} \text{standard} & \text{if } Z^THZ \text{ is positive definite;} \\ \text{nonstandard} & \text{if } Z^THZ \text{ is not positive definite;} \\ \text{intermediate} & \text{if } x \text{ is not a minimizer.} \end{cases}$$

In each case, the term requires a specification of A, which is omitted only when its meaning is obvious. We stress that the same point can be, for example, a minimizer with respect to one working set A, but intermediate with respect to another (usually, A with one or more constraints deleted).

We now examine the properties of intermediate iterates that occur *after* a constraint is deleted at one minimizer, but *before* the next minimizer is reached. Each such iterate is associated with a unique *most recently deleted constraint*. Consider a sequence of *consecutive* intermediate iterates $\{x_k\}$, $k = 0, \ldots, N$, with the following three features:

I1. x_k is intermediate with respect to the working set A_k;

I2. A_0 is obtained by deleting the constraint with normal a_* from the working set A_*, so that x_0 is a minimizer with respect to A_*;

I3. x_k, $1 \le k \le N$, is *not a minimizer* with respect to any valid working set.

At x_k, p_k is defined using A_k as A (and, if necessary, a_* as a_s) in (4.1), (4.2), (4.3) or (4.4). (Note that (4.4) may be used only at x_0.)

Let Z_* denote a basis for the null space of A_*. For purposes of this discussion, the position of a_*^T in A_* is irrelevant, and hence we assume that A_* has the form

$$A_* = \begin{pmatrix} A_0 \\ a_*^T \end{pmatrix}. \tag{4.6}$$

Because of the inertia-controlling strategy, the reduced Hessian $Z_*^T H Z_*$ must be positive definite. Relation (4.6) implies that

$$p^T H p > 0 \text{ for any nonzero } p \text{ such that } A_0 p = 0 \text{ and } a_*^T p = 0. \tag{4.7}$$

If the iterate following x_k is intermediate and the algorithm continues, α_k is the step to the nearest constraint, and *a constraint is added to the working set* at each x_k, $k \geq 1$. If a constraint is added and x_k is standard, *it must hold that* $\alpha_k < 1$. (Otherwise, if $\alpha_k = 1$, $x_k + p_k$ is a minimizer with respect to A_k, and the sequence of intermediate iterates ends.) Let a_k denote the normal of the constraint added to A_k at x_{k+1} to produce A_{k+1}, so that the form of A_{k+1} is

$$A_{k+1} = \begin{pmatrix} A_k \\ a_k^T \end{pmatrix} = \begin{pmatrix} A_0 \\ a_0^T \\ \vdots \\ a_k^T \end{pmatrix}. \tag{4.8}$$

We now prove several lemmas leading to the result that the gradient at each intermediate iterate x_k may be expressed as a linear combination of A_k and a_*. For simplicity, whenever possible we adopt the convention that unbarred and barred quantities are associated with intermediate iterates k and $k+1$ respectively.

LEMMA 4.1. *Let g and A denote the gradient and working set at an intermediate iterate x where p is defined by* (4.1)–(4.3)*, and a_* is the most recently deleted constraint. Let $\bar{x} = x + \alpha p$, and assume that constraint a is added to A at $\bar{x}$, giving the working set $\bar{A}$. If there exist a vector v and a scalar v_* such that*

$$g = A^T v - v_* a_*, \quad \textit{with} \quad v_* > 0, \tag{4.9}$$

then

(a) *$\bar{g}$, the gradient at $\bar{x}$, is also a linear combination of A^T and a_*;*

(b) *there exist a vector $\bar{v}$ and scalar $\bar{v}_*$ such that*

$$\bar{g} = \bar{A}^T \bar{v} - \bar{v}_* a_*, \quad \textit{with} \quad \bar{v}_* > 0. \tag{4.10}$$

Proof. Because φ is quadratic,

$$g(x + \alpha p) = g + \alpha H p. \tag{4.11}$$

We now consider the form of $\bar{g}$ for the three possible definitions of p.

When the reduced Hessian is positive definite, p satisfies $g + Hp = A^T \mu$, so that $Hp = -g + A^T \mu$. Substituting from this expression and (4.9) in (4.11), we obtain (a) from

$$\bar{g} = g + \alpha H p = (1-\alpha) g + \alpha A^T \mu = A^T \lambda - \bar{v}_* a_*,$$

where $\lambda = (1-\alpha) v + \alpha \mu$ and $\bar{v}_* = (1-\alpha) v_*$. Since $\alpha < 1$, (b) is obtained by forming $\bar{v}$ from λ and a zero component corresponding to row a^T in $\bar{A}$.

When the reduced Hessian is singular, p is defined as $\beta\hat{p}$, where $\beta \neq 0$ and $\hat{p}$ satisfies (4.2), so that $Hp = -\beta A^T\nu$. Substituting from this relation and (4.9) in (4.11) gives

$$\bar{g} = g + \alpha Hp = g - \alpha\beta A^T\nu = A^T(v - \alpha\beta\nu) - v_* a_*,$$

and (4.10) holds with $\bar{v}_* = v_*$ and $\bar{v}$ formed by augmenting $\lambda = v - \alpha\beta\nu$ with a zero component as above.

Finally, when the reduced Hessian is indefinite and the search direction is defined by (4.3), $Hp = g - A^T\mu$. Substituting from this relation and (4.9) in (4.11), we obtain

$$\begin{aligned}\bar{g} &= g + \alpha Hp = g + \alpha(g - A^T\mu)\\ &= (1+\alpha)g - \alpha A^T\mu\\ &= (1+\alpha)A^Tv - \alpha A^T\mu - (1+\alpha)v_* a_*\\ &= A^T\lambda - \bar{v}_* a_*,\end{aligned}$$

where $\lambda = (1+\alpha)v - \alpha\mu$ and $\bar{v}_* = (1+\alpha)v_*$. Since $v_* > 0$, $\bar{v}_*$ must be positive, and $\bar{g}$ has the desired form. ☐

To begin the induction, note that if the multiplier associated with a_* at x_0 is *negative*, then, from (4.6),

$$g_0 = A_*^T\mu = A_0^T\mu_0 - v_*^0 a_*, \tag{4.12}$$

where $v_*^0 = -\mu_* > 0$. The next lemma treats the other possibility, that a *zero* multiplier was associated with a_*, i.e., that x_0 is a stationary point with respect to A_0. The situation is possible only if the reduced Hessian associated with A_0 is indefinite. (If it were positive definite, the algorithm would delete further constraints; if it were singular, the algorithm would terminate at x_0.)

LEMMA 4.2. *Assume that the reduced Hessian is indefinite at the first intermediate iterate x_0, and that a zero multiplier is associated with a_*. Then*

$$g_0^Tp_0 = 0, \quad p_0^THp_0 < 0 \quad \text{and} \quad a_*^Tp_0 > 0. \tag{4.13}$$

If $\alpha_0 > 0$, then $g_1 = g(x_0 + \alpha_0 p_0)$ may be written as a linear combination of a_* and the rows of A_0. Moreover, there exist a vector v^1 and scalar v_*^1 such that

$$g_1 = A_1^Tv^1 - v_*^1 a_*, \tag{4.14}$$

with $v_*^1 > 0$.

Proof. Since a zero multiplier is associated with a_*, x_0 is a stationary point with respect to A_0, i.e., $g_0 = A_0^T\mu_0$. Multiplying by p_0^T shows that $p_0^Tg_0 = 0$. Using (4.5), p_0 satisfies

$$Hp_0 = -A_0^Tw_0 - w_* a_*, \tag{4.15}$$

where $w_* > 0$, so that

$$p_0^THp_0 = -w_* a_*^Tp_0. \tag{4.16}$$

Rewriting the definition (4.5) of p as

$$\begin{pmatrix} H & A_0^T \\ A_0 & \end{pmatrix}\begin{pmatrix} p_0 \\ w_0 \end{pmatrix} = w_*\begin{pmatrix} -a_* \\ 0 \end{pmatrix} \quad \text{with } w_* > 0, \tag{4.17}$$

Lemma 3.8 implies that $a_*^T p_0 > 0$. It then follows from (4.16) that $p_0^T H p_0 < 0$, which completes verification of (4.13).

Now we assume that $\alpha_0 > 0$. Since $g_1 = g_0 + \alpha_0 H p_0$, (4.15) and the relation $g_0 = A_0^T \mu_0$ give

$$g_1 = g_0 + \alpha_0 H p_0 = A_0^T \mu_0 - \alpha_0 A_0^T w_0 - \alpha_0 w_* a_* = A_0^T \lambda - v_*^1 a_*,$$

where $v_*^1 = \alpha_0 w_*$ and $\lambda = \mu_0 - \alpha_0 w_0$. Since $\alpha_0 > 0$ and $w_* > 0$, v_*^1 is strictly positive, and g_1 has the desired form. If constraint a_0 is added to the working set at the new iterate, g_1 can equivalently be written as in (4.14) by forming v^1 from an augmented version of λ as in Lemma 4.1. ☐

We are now able to derive some useful results concerning the sequence of intermediate iterates.

LEMMA 4.3. *Given a sequence of consecutive intermediate iterates* $\{x_k\}$ *satisfying properties* I1–I3*, the gradient* g_k *satisfies* (4.9) *for* $k \geq 0$ *if a constraint with a negative multiplier is deleted at* x_0*, and for* $k \geq 1$ *if a constraint with a zero multiplier is deleted at* x_0 *and* $\alpha_0 > 0$*.*

Proof. If a constraint with a negative multiplier is deleted at x_0, (4.9) holds at x_0 by definition (see (4.12)). If a constraint with a zero multiplier is deleted at x_0 and $\alpha_0 > 0$, Lemma 4.2 shows that (4.9) holds at x_1. Lemma 4.1 therefore applies at all subsequent intermediate iterates, where we adopt the convention that v increases in dimension by one at each step to reflect the fact that A_k has one more row than A_{k-1}. ☐

LEMMA 4.4. *Let* $\{x_k\}$ *be a sequence of consecutive intermediate iterates satisfying properties* I1–I3*. Given any vector* p *such that* $A_k p = 0$*, the following two properties hold for* $k \geq 0$ *if a constraint with a negative multiplier is deleted at* x_0*, and for* $k \geq 1$ *if a constraint with a zero multiplier is deleted at* x_0 *and* $\alpha_0 > 0$*:*

(a) *if* $g_k^T p < 0$*, then* $a_*^T p > 0$*;*
(b) *if* $a_*^T p > 0$*, then* $g_k^T p < 0$*.*

Proof. We know from part (b) of Lemma 4.3 that, for the stated values of k, there exist a vector v^k and positive scalar v_*^k such that

$$g_k = A_k^T v^k - v_*^k a_*.$$

Therefore, $g_k^T p = -v_*^k a_*^T p$ and the desired results are immediate. ☐

LEMMA 4.5. *Assume that* $\{x_k\}$, $k = 0, \ldots, N$*, is a sequence of consecutive intermediate iterates satisfying* I1–I3*, where each* x_k*,* $1 \leq k \leq N$*, is not a stationary point with respect to* A_k*. Assume further that* $\alpha_0 > 0$ *if a zero multiplier is deleted at* x_0*, and that* α_N *is the step to the constraint with normal* a_N*, which is added to* A_N *to form the working set* A_{N+1}*. Let* $x_{N+1} = x_N + \alpha_N p_N$*.*

(a) *If* x_{N+1} *is a stationary point with respect to* A_{N+1}*, then* a_N *is linearly dependent on* A_N^T *and* a_**, and* $Z_{N+1}^T H Z_{N+1}$ *is positive definite;*
(b) *If* a_N *is linearly dependent on* A_N^T *and* a_**, then* x_{N+1} *is a minimizer with respect to* A_{N+1}*.*

Proof. By construction, the working set A_* has full row rank, so that a_* is linearly independent of the rows of A_0. We know from part (b) of Lemma 4.3 that

$$g_k = A_k^T v^k - v_*^k a_*, \quad k = 1, \ldots, N, \tag{4.18}$$

where $v_*^k > 0$. Since we have assumed that x_k is not a stationary point with respect to A_k for any $1 \leq k \leq N$, (4.18) shows that a_*^T is linearly independent of A_k.

Furthermore, part (a) of Lemma 4.1 implies that there exists a vector λ such that

$$g_{N+1} = A_N^T\lambda - v_*^{N+1}a_*, \tag{4.19}$$

where $v_*^{N+1} > 0$. It follows from the linear independence of a_*^T and A_N that x_{N+1} cannot be a stationary point with respect to the "old" working set A_N.

To show part (a), assume that x_{N+1} is a stationary point with respect to A_{N+1} (which includes a_N), i.e.,

$$g_{N+1} = A_N^T\mu + \mu_N a_N, \tag{4.20}$$

where μ_N (the multiplier associated with a_N) must be nonzero. Equating the right-hand sides of (4.19) and (4.20), we obtain

$$A_N^T\lambda - v_*^{N+1}a_* = A_N^T\mu + \mu_N a_N. \tag{4.21}$$

Since $v_*^{N+1} \neq 0$ and $\mu_N \neq 0$, this expression implies that we may express a_* as a linear combination of A_N^T and a_N, where the coefficient of a_N is *nonzero*:

$$a_* = A_N^T\xi + \gamma a_N, \quad \text{with} \quad \gamma = -\frac{\mu_N}{v_*^{N+1}} \neq 0 \tag{4.22}$$

and $\xi = (1/v_*^{N+1})(\lambda - \mu)$.

Stationarity of x_{N+1} with respect to A_{N+1} thus implies a *special relationship* among the most recently deleted constraint, the working set at x_N and the newly encountered constraint. Any nonzero vector p in the null space of A_{N+1} satisfies

$$A_{N+1}p = \begin{pmatrix} A_N \\ a_N^T \end{pmatrix} p = 0. \tag{4.23}$$

For any such p, it follows from the structure of A_{N+1} (see (4.8)) that $A_0p = 0$, and from (4.22) that $a_*^Tp = 0$; hence, p lies in the null space of A_*. Since $Z_*^THZ_*$ is positive definite (i.e., (4.7) holds), we conclude that $p^THp > 0$ for p satisfying (4.23). Thus, the reduced Hessian at x_{N+1} with respect to A_{N+1} is *positive definite*, and x_{N+1} is a minimizer with respect to A_{N+1}.

To verify part (b), assume that a_N is linearly dependent on A_N and a_*, i.e., that $a_N = A_N^T\beta + a_*\beta_*$, where $\beta_* \neq 0$. Simple rearrangement then gives $a_* = (1/\beta_*)a_N - (1/\beta_*)A_N^T\beta$. Substituting in (4.19), we obtain

$$g_{N+1} = A_N^T\lambda - \frac{v_*^{N+1}}{\beta_*}\, a_N - \frac{v_*^{N+1}}{\beta_*}\, A_N^T\beta,$$

which shows that x_{N+1} must be a stationary point with respect to A_{N+1}. Positive-definiteness of the reduced Hessian follows as before, and hence x_{N+1} is a minimizer with respect to A_{N+1}. ☐

Lemma 4.5 is crucial in ensuring that *adding* a constraint in an inertia-controlling algorithm cannot produce a stationary point where the reduced Hessian is not positive definite.

4.3. Properties of the search direction. When the reduced Hessian is positive definite, it is straightforward to show that the search direction possesses the feasibility and descent properties discussed in Section 2.3.

THEOREM 4.6. *Consider an iterate x and a valid working set A such that Z^THZ is positive definite. If p as defined by* (4.1) *is nonzero, then p is a descent direction.*

Furthermore, if constraint a_* *is the most recently deleted constraint, it also holds that* $a_*^T p > 0$.

Proof. See Fletcher [13, page 89]. Writing out the equations of (4.1), we have

$$g + Hp = A^T\mu \quad \text{and} \quad Ap = 0.$$

Multiplying the first equation by p^T gives $g^Tp = -p^THp$. Since $p = Zp_Z$ for some nonzero p_Z and Z^THZ is positive definite, p^THp must be strictly positive, and hence $g^Tp < 0$. If constraint a_* is the most recently deleted constraint, x must be part of a sequence of intermediate iterates satisfying properties I1–I3 (Section 4.2), where a negative multiplier was deleted at the first point of the sequence. Lemma 4.4 thus shows that $a_*^Tp > 0$. □

We now wish to verify that the search direction at a nonstandard iterate (which must be intermediate) possesses the desired properties. Lemma 4.2 shows that p is a direction of negative curvature when a constraint with a zero multiplier has just been deleted. The following theorems treat the two possible situations when the most recently deleted constraint has a negative multiplier.

THEOREM 4.7. *When the reduced Hessian is singular at a nonstandard iterate* x*, the search direction is a descent direction of zero curvature. If* a_* *is the most recently deleted constraint, it also holds that* $a_*^Tp > 0$.

Proof. When Z^THZ is singular, p is defined by (4.2) and hence satisfies $Hp = -\beta A^T\nu$. Multiplying this relation by p^T, we obtain $p^THp = 0$, which verifies that p is a direction of zero curvature. A nonstandard iterate x must be part of a sequence of intermediate iterates satisfying properties I1–I3. We know from Lemma 4.5 that any such x cannot be a stationary point, and hence $g^Tp \neq 0$. Thus, the sign of β can always be chosen so that $g^Tp < 0$. Lemma 4.4 then implies that $a_*^Tp > 0$, where a_* is the normal of the most recently deleted constraint. □

THEOREM 4.8. *When the reduced Hessian is indefinite at a nonstandard iterate and the search direction is defined by (4.3),* p *is a descent direction of negative curvature. If* a_* *is the most recently deleted constraint, it also holds that* $a_*^Tp > 0$.

Proof. Since p satisfies $Hp + A^T\mu = g$ and $Ap = 0$, it follows that

$$p^THp = g^Tp. \tag{4.24}$$

As in Theorem 4.7, x must be part of a sequence of intermediate iterates satisfying properties I1–I3. Furthermore, Lemma 4.3 shows that

$$g = A^Tv - v_*a_*, \quad \text{with} \quad v_* > 0,$$

where a_* is the normal of the most recently deleted constraint. Substituting for g in (4.3) and rearranging, we see that p satisfies

$$K\begin{pmatrix} p \\ w \end{pmatrix} = \begin{pmatrix} H & A^T \\ A & \end{pmatrix}\begin{pmatrix} p \\ w \end{pmatrix} = v_*\begin{pmatrix} -a_* \\ 0 \end{pmatrix},$$

and it follows from Lemma 3.8 that $a_*^Tp > 0$. This property implies first (from Lemma 4.4) that $g^Tp < 0$, and then (from (4.24)) that $p^THp < 0$ as required. □

If $\alpha_F = 1$ at a standard iterate, a constraint is not added to the working set at the next iterate, which is automatically a minimizer with respect to the same working set (see the logic for constraint addition in Figure 2). If a new iterate *happens* to be a stationary point under any other circumstances, we now show that the multiplier corresponding to the newly added constraint must be strictly positive.

LEMMA 4.9. *Assume that x is a typical intermediate iterate, with associated working set A, under the same conditions as in* Lemma 4.5. *Let $\bar{x} = x + \alpha p$, where $\alpha > 0$ and constraint a is added to the working set at $\bar{x}$, and let $\bar{A}$ denote the new working set. If $\bar{x}$ is a stationary point with respect to $\bar{A}$, then the Lagrange multiplier associated with the newly added constraint is positive.*

Proof. If $\bar{x}$ is a stationary point with respect to $\bar{A}$, we have by definition that $\bar{g} = A^T\mu_A + a\mu_a$, where μ_a is the multiplier corresponding to the newly added constraint. Since the conditions of this lemma are the same as those of Lemma 4.5,

$$-v_* a_* = A^T\lambda + \mu_a a, \quad \text{where} \quad v_* > 0 \tag{4.25}$$

(see (4.21)). Lemma 4.2 and Theorems 4.6–4.8 show that $a_*^T p > 0$ at every intermediate iterate. Since constraint a is added to the working set, we know that $a^T p < 0$. Relation (4.25) shows that $-v_* a_*^T p = \mu_a a^T p$, and we conclude that $\mu_a > 0$ as desired. ∎

4.4. Choosing the initial working set. Inertia-controlling methods require a procedure for finding an initial working set A_0 that has full row rank and an associated positive-definite reduced Hessian $Z_0^T H Z_0$. Two different inertia-controlling methods starting with the same working set A_0 will generate identical iterates. However, procedures for finding A_0 are usually dependent on the method used to solve the KKT system and therefore A_0 may vary substantially from one method to another. Ironically, this implies that different inertia-controlling methods seldom generate the same iterates in practice!

In order to ensure that the reduced Hessian is positive definite, the initial working set may need to include "new" constraints that are not specified in the original problem. These have been called *temporary constraints*, *pseudo-constraints* (Fletcher and Jackson [16]), or *artificial constraints* (Gill and Murray [18]). The only requirement for a temporary constraint is linear independence from constraints already in the working set. The strategy for choosing temporary constraints depends on the mechanics of the particular QP method.

Simple bounds involving the current values of variables are convenient temporary constraints in certain contexts (see, e.g., Fletcher and Jackson [16]). For example, suppose that the value of the first variable at the initial point is 6. The two "opposite" temporary constraints $x_1 \geq 6$ and $-x_1 \geq -6$ (equivalently, $x_1 \leq 6$) are clearly active at the initial point. The first coordinate vector e_1 is thus a candidate for inclusion in the initial working set if it satisfies the linear independence criterion. If such a bound is included, the first variable is "temporarily" fixed at 6. The sign of the temporary constraint normal does not affect the null space of the working set, and hence is *irrelevant* until a minimizer is reached. At a minimizer, the sign of each temporary constraint normal is chosen to make its multiplier nonpositive, so that the constraint may be deleted. Temporary constraints are usually deleted first if there is a choice.

Since a reduced Hessian of dimension zero is positive definite, the strategy originally associated with inertia-controlling methods was *always* to start at a "temporary vertex", i.e., to choose an initial working set of n constraints, regardless of the nature of the reduced Hessian (see Fletcher [12] and Gill and Murray [18]). However, this approach may be inefficient because of the nontrivial effort that must be expended to delete all the temporary constraints, and has been superseded by more sophisticated strategies.

Ideally, the initial working set should be well conditioned and contain as few temporary constraints as possible. A strategy that attempts to fulfill these aims is used in the method of QPSOL [22, 19]. Let A' denote the subset of rows of $\mathcal{A}$ corresponding to the set of constraints active at x_0. A trial working set (the maximal linearly independent subset of the rows of A') is selected by computing an orthogonal-triangular factorization in which one row is added at a time. If the diagonal of the triangular factor resulting from addition of a particular constraint is "too small", the constraint is considered dependent and is not included.

Let A_W denote the resulting trial working set, with Z_W a null-space basis for A_W. If $Z_W^T H Z_W$ is positive definite, A_W is an acceptable initial working set, and A_0 is taken as A_W. Otherwise, the requisite temporary constraint normals are taken as the columns of Z_W that lie in the subspace spanned by the eigenvectors associated with the nonpositive eigenvalues of $Z_W^T H Z_W$. With the TQ factorization (see (6.1)), these columns can be identified by attempting to compute the Cholesky factorization of $Z_W^T H Z_W$ *with symmetric interchanges* (for details, see Gill et al. [19]).

In contrast, methods that rely on sparse factorizations to solve KKT-related systems explicitly (see Section 5.1) have more difficulty in defining A_0 efficiently, since there is no guaranteed technique for minimizing the number of temporary constraints. "Crash" procedures for choosing the initial working set in the context of sparse QP are described in [29].

The task of finding A_0 is also complicated in practice by the desirability of specifying a "target" initial working set. For example, the QP may occur as a subproblem within an SQP method for nonlinearly constrained optimization with a "warm start" option; see Gill *et al.* [23].

4.5. Convergence. In all our discussion thus far, we have assumed at various crucial junctures that $\alpha > 0$, because of the theoretical (and practical) difficulties in treating degenerate stationary points. A *degenerate stationary point* for (1.1) is a point at which the gradient of φ is a linear combination of the active constraint normals, but the active constraint normals are linearly dependent. A degenerate vertex is the most familiar example of such a point.

At a stationary point, progress can be made only by deleting a constraint. If the resulting search direction immediately "hits" an idle constraint, the algorithm is forced to take a zero step and add a constraint to the working set *without moving*. This situation cannot continue indefinitely if the active constraints are linearly independent. When the active constraints are linearly dependent, however, *cycling* (a non-terminating sequence of working sets) may occur if "standard" choices are made for the constraints to be deleted and added.

Practical techniques for moving away from degenerate stationary points in both linear and quadratic programming are discussed in, for example, Fletcher [15, 14], Busovača [4], Dax [11], Osborne [35], Ryan and Osborne [37], Gill et al. [24] and Gould [29].

Proofs of convergence for inertia-controlling methods if no degenerate stationary points exist have been given in [12, 14, 18, 28]. We therefore simply state the result.

THEOREM 4.10. *If $\varphi(x)$ is bounded below in the feasible region of (1.1) and the feasible region contains no degenerate stationary points, an inertia-controlling algorithm converges in a finite number of iterations to a point x where*

1. $Z^T g = 0$, $Z^T H Z$ *is positive definite and* $\mu > 0$*; or*
2. $Z^T g = 0$, $Z^T H Z$ *is singular and* $\mu \geq 0$. □

5. The Formulation of Algorithms. Given the same initial working set, inertia-controlling methods generate mathematically identical iterates. Practical inertia-controlling methods differ in the techniques used to determine the nature of the reduced Hessian and to compute the search direction and Lagrange multipliers.

5.1. Using a nonsingular extended KKT system. When solving a general QP with an inertia-controlling method, the "real" KKT matrix (i.e., the KKT matrix including the current working set) may be singular for any number of iterations. In this section, we show how to define the vectors of interest in terms of linear systems involving a *nonsingular* matrix that (optionally) includes the normal of the most recently deleted constraint—in effect, an "extended" KKT matrix. Fletcher's original method [12] uses the approach to be described, although he describes the computations in terms of a partitioned inverse. Any "black box" equation solver that provides the necessary information may be used to solve these equations (see, e.g., Gould [29] and Gill et al. [25]).

At a given iterate, let A_* denote either the current working set A or a matrix of *full row rank* whose i_*-th row is a_* (the most recently deleted constraint) and whose remaining rows are those of A. (If $A_* = A$, i_* is taken as zero.) The row dimension of A_* is denoted by m_*, which is m when $A_* = A$ and $m+1$ when $A_* \neq A$. Let Z and Z_* be null-space bases for A and A_*. The inertia-controlling strategy guarantees that the reduced Hessian $Z_*^T H Z_*$ is positive definite. We allow A_* to be A *only when* $Z^T HZ$ *is positive definite*, in order to guarantee its nonsingularity at intermediate iterates. (Recall that $Z^T HZ$ can change from indefinite to singular following a constraint addition.) However, it may be convenient to retain a_* in A_* even in the positive-definite case.

The matrix K_* is defined as

$$K_* = \begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix}, \tag{5.1}$$

and we emphasize that K_* must be nonsingular (see Corollary 3.2). Let u, v, y and w be the (unique) solutions of

$$\begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{5.2}$$

$$\begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix} \begin{pmatrix} y \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ e_* \end{pmatrix}, \tag{5.3}$$

where u and y have n components, v and w have m_* components, and e_* denotes the i_*-th coordinate vector of dimension m_*. When $K_* = K$, y and w may be taken as zero. Any vector name with subscript "A" denotes the subvector corresponding to columns of A^T, and similarly for the subscript "$*$". If $i_* = 0$, the i_*-th component of a vector is null.

The vectors q and μ associated with the KKT system (2.7) satisfy

$$Hq + A^T\mu = g, \quad Aq = 0, \tag{5.4}$$

so that $q = u$ of (5.2) when $K = K_*$. In an inertia-controlling method, the search direction p is taken as $-q$ in the positive-definite case (see (4.1)), as y in the singular case or in the indefinite case with a zero multiplier (see (4.2) and (4.4)), or as q (see (4.3)). Thus, p is available directly from (5.2) or (5.3) in two situations: when $K_* = K$, in which case p must be $-q$ since $Z^T HZ$ is positive definite; or when $p = y$. The next lemma shows how to obtain q and μ from the vectors of (5.2) and (5.3) when $K_* \neq K$.

LEMMA 5.1. *If K is nonsingular and $K_* \neq K$, the vectors q and μ are given by*

$$q = u + \beta y$$
$$\mu = v_A + \beta w_A, \quad \textit{where} \quad \beta = -v_*/w_*. \tag{5.5}$$

Proof. Writing out the equations of (5.2) and (5.3), we have

$$Hu + A^T v_A + a_* v_* = g, \quad Au = 0, \quad a_*^T u = 0;$$
$$Hy + A^T w_A + a_* w_* = 0, \quad Ay = 0, \quad a_*^T y = 1.$$

For any scalar β, the vectors $u' = u + \beta y$ and $v' = v + \beta w$ satisfy

$$Hu' + A^T v'_A + a_*(v_* + \beta w_*) = g \quad \text{and} \quad A^T u' = 0. \tag{5.6}$$

Both K and K_* are nonsingular, which implies that $w_* \neq 0$ (see Corollary 3.7). If β is chosen as $-v_*/w_*$, the coefficient of a_* in (5.6) is zero, and u' and v'_A satisfy (5.4). The desired result follows from the uniqueness of q and μ. □

When $K_* \neq K$, the following two lemmas indicate how to use u, v, y and w to decide on the status of the reduced Hessian and of the current iterate.

LEMMA 5.2. *Assume that $K_* \neq K$. Then:* (a) *if $w_* < 0$, $Z^T HZ$ is positive definite;* (b) *if $w_* = 0$, $Z^T HZ$ is singular; and* (c) *if $w_* > 0$, $Z^T HZ$ is indefinite.*

Proof. Since A_* is chosen so that $Z_*^T H Z_*$ is positive definite, the results follow from Corollary 3.7. □

LEMMA 5.3. *Assume that $K_* \neq K$. The point x is a stationary point with respect to A if $u = 0$ and $v_* = 0$.*

Proof. The result is immediate from the definition of u and v. □

5.2. Updating the required vectors. The next four lemmas specify how u, v, y and w can be *recurred* from iteration to iteration. Note that "old" and "new" versions of u and y always have n components.

LEMMA 5.4 (Move to a new iterate). *Suppose that x is an iterate of an inertia-controlling method. Let $\bar{x} = x + \alpha p$. The solution of*

$$\begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} = \begin{pmatrix} \bar{g} \\ 0 \end{pmatrix}, \tag{5.7}$$

where $\bar{g} = g(\bar{x}) = g + \alpha Hp$, is given by

$$\bar{u} = \begin{cases} (1-\alpha)u \\ (1+\alpha)u \\ u \end{cases}, \qquad \bar{v}_* = \begin{cases} (1-\alpha)v_* & \textit{if } p = -q \\ (1+\alpha)v_* & \textit{if } p = q \\ v_* - \alpha w_* & \textit{if } p = y \end{cases}$$
$$\bar{v}_A = v_A - \alpha(a_*^T p) w_A. \tag{5.8}$$

Proof. In this lemma, the move from x to $\bar{x}$ changes only the gradient (not the working set). The desired result can be verified by substitution from Lemma 5.1 and the various definitions of p. □

Following the addition of a constraint (say, a) to the working set, the "real" reduced Hessian may become positive definite, so that strictly speaking a_* is no longer necessary. Nonetheless, it may be desirable to retain a_* in A_* for numerical reasons;

various strategies for making this decision are discussed in [12]. Updates can be performed in either case, using the n-vector z and m_*-vector t defined by

$$\begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix} \begin{pmatrix} z \\ t \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}, \tag{5.9}$$

i.e., such that

$$Hz + A^T t_A + t_* a_* = a, \quad Az = 0 \quad \text{and} \quad a_*^T z = 0. \tag{5.10}$$

We first consider the case when a can be added directly to A_*. Following the updates given in the next lemma, m_* increases by one and the "new" v and w have one additional component.

LEMMA 5.5 (Constraint addition; independent case). *Let x denote an iterate of an inertia-controlling method. Assume that constraint a is to be added to the working set at x, where A_*^T and a are linearly independent. Let*

$$\rho = \frac{a^T u}{a^T z} \quad \text{and} \quad \eta = \frac{a^T y}{a^T z}. \tag{5.11}$$

Then the vectors $\bar u$, $\bar v$, $\bar y$ and $\bar w$ defined by

$$\begin{aligned} \bar u &= u - \rho z, & \bar v &= \begin{pmatrix} v - \rho t \\ \rho \end{pmatrix} \\ \bar y &= y - \eta z, & \bar w &= \begin{pmatrix} w - \eta t \\ \eta \end{pmatrix} \end{aligned} \tag{5.12}$$

satisfy

$$\begin{pmatrix} H & A_*^T & a \\ A_* & & \\ a^T & & \end{pmatrix} \begin{pmatrix} \bar u \\ \bar v \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \quad \begin{pmatrix} H & A_*^T & a \\ A_* & & \\ a^T & & \end{pmatrix} \begin{pmatrix} \bar y \\ \bar w \end{pmatrix} = \begin{pmatrix} 0 \\ e_* \\ 0 \end{pmatrix}. \tag{5.13}$$

If $A_* = A$, y and w have dimension zero, and are not updated.

Proof. When a and A_*^T are linearly independent, (5.10) shows that z must be nonzero. Since $A_* z = 0$ and $Z_*^T H Z_*$ is positive definite, $a^T z = z^T H z > 0$, so that ρ and η are well defined.

For any scalar ρ, (5.2) and (5.10) imply that

$$\begin{pmatrix} H & A_*^T & a \\ A_* & & \\ a^T & & \end{pmatrix} \begin{pmatrix} u - \rho z \\ v - \rho t \\ \rho \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ a^T u - \rho a^T z \end{pmatrix}. \tag{5.14}$$

The linear independence of a and A_*^T means that the solution vectors of (5.13) are unique. By choosing ρ so that the last component of the right-hand side of (5.14) vanishes, we see that $\bar u$ and $\bar v$ of (5.12) satisfy the first equation of (5.13). A similar argument gives the updates for $\bar y$ and $\bar w$. ☐

If $Z^T H Z$ is positive definite and $K_* \neq K$, a_* can be deleted from A_*, and K_* then becomes K itself. The following lemma may be applied in two situations: when a constraint is deleted from the working set at a minimizer and the reduced Hessian remains positive definite after deletion; and at an intermediate iterate after a constraint has been added that makes $Z^T H Z$ positive definite.

LEMMA 5.6 (Deleting a_* from A_*). *Suppose that: x is an iterate of an inertia-controlling method, $K_* \neq K$, and Z^THZ is positive definite. Then the vectors $\bar{u}$ and $\bar{v}$ defined by*

$$\bar{u} = u + \zeta y, \quad \bar{v}_A = v_A + \zeta w_A, \quad \text{where} \quad \zeta = -\frac{v_*}{w_*}, \tag{5.15}$$

satisfy

$$H\bar{u} + A^T\bar{v} = g, \quad A\bar{u} = 0. \tag{5.16}$$

Proof. Let $u' = u + \zeta y$, $v' = v + \zeta w$ for some scalar ζ. Substituting these values in (5.2), we have

$$H(u + \zeta y) + A^T(v_A + \zeta w_A) + a_*(v_* + \zeta w_*) = g.$$

It follows that (5.16) will be satisfied by u' and v'_A if $v_* + \zeta w_* = 0$. It is permissible to delete a_* from A_* only if Z^THZ is positive definite, which means that $w_* < 0$, and hence ζ is well defined. □

Note that y and w are no longer needed to define the search direction after a_* has been removed.

The only remaining possibility occurs when a, the constraint to be added, is linearly dependent on A_*^T; in this case, $z = 0$ in (5.9). We know from Lemma 4.5 that the iterate just reached must be a *minimizer with respect to the working set composed of A^T and a*, which means that a_* is no longer necessary. However, it is not possible to update u using Lemma 5.5 (because $a^Tz = 0$), nor to apply Lemma 5.6 (because w_* may be zero). The following lemma gives an update that simultaneously removes a_* from A_* and adds a to the working set. After application of these updates, $\bar{A}$ is the "real" working set at $\bar{x}$, and the algorithm either terminates or deletes a constraint (which cannot be a; see Lemma 4.9).

LEMMA 5.7 (Constraint addition; dependent case). *Suppose that x is an iterate of an inertia-controlling method and that $K_* \neq K$. Assume that a is to be added to the working set at x, and that a and A_*^T are linearly dependent. Let $\bar{A}$ denote A with a^T as an additional row, and define $\omega = v_*/t_*$. The vectors $\bar{u}$ and $\bar{v}$ specified by*

$$\bar{u} = 0, \quad \bar{v}_A = v_A - \omega t_A, \quad \bar{v}_a = \omega, \tag{5.17}$$

where $\bar{v}_a$ denotes the component of $\bar{v}$ corresponding to a, satisfy

$$\begin{pmatrix} H & \bar{A}^T \\ \bar{A} & \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{5.18}$$

Proof. First, observe that linear dependence of A_*^T and a means that $z = 0$. Lemma 2.1 shows that a cannot be linearly dependent on A^T, which implies that $t_* \neq 0$. Lemma 4.5 tells us that x must be a minimizer with respect to a working set, so that $\bar{u} = 0$. The desired results follow from substitution. □

The algorithmic implications of these lemmas are summarized in the following theorem. The ability to recur the required vectors has previously been proved only under the assumption that the initial point is a minimizer (see Lemma 5.9).

THEOREM 5.8. *In an inertia-controlling method based on using a nonsingular matrix K_* as described, the linear system* (5.2) *needs to be solved explicitly for u and*

v only once (at the first iterate); these vectors can thereafter be updated. The vectors y and w must be computed by solving (5.3) *at each minimizer, since w is used to determine the nature of the reduced Hessian when a constraint is deleted; y and w may be updated when a constraint is added to the working set. The vectors z and t must be computed by solving* (5.9) *whenever a constraint is added to the working set.* ∎

Figures 4 and 5 specify the computations associated with deleting and adding a constraint (the boxed portions of Figures 1 and 2).

$a_* \leftarrow a_s$; $A_* \leftarrow A$;
compute y and w by solving (5.3);
determine the nature of Z^THZ (Lemma 5.2);
if *positive_definite* **then**
  (optionally) delete a_* from A_*;
  update u and v (Lemma 5.6);
end if

Figure 4. Deleting constraint a_s from the working set.

solve (5.9) to obtain z and t;
if $z \neq 0$ **then**
  add a to A_*; update u, v, y and w (Lemma 5.5);
  determine the nature of Z^THZ (Lemma 5.2);
  if *positive_definite* **then**
    (optionally) delete a_* from A_*;
    update u and v (Lemma 5.6);
  end if
else $(z = 0)$
  remove a_* from A_* and add a to the working set;
  update u and v (Lemma 5.7);
  positive_definite $\leftarrow$ **true**;
end if

Figure 5. Adding constraint a to the working set.

For simplicity, two special circumstances are not shown: in Figure 4, a_* is always deleted from A_* when $\mu_* = 0$ and the reduced Hessian remains positive definite after deletion, to allow the algorithm to proceed if another constraint is deleted; and if $A_* = A$ in Figure 5, it is not necessary to test the nature of the reduced Hessian, which must be positive definite.

A final lemma indicates a further efficiency that may be achieved once a minimizer has been reached.

Lemma 5.9. *If an iterate x is a minimizer with respect to A, the vector u is zero for all subsequent iterations.*

Proof. When x is a minimizer with respect to a working set A, g is a linear combination of the columns of A^T, so that $u = 0$. The result of the lemma follows by noting that none of the recurrence relations for u alters this value. Hence, only v, y and w need to be stored and updated thereafter. ∎

6. Two Specific Methods. In this section we give details concerning two specific inertia-controlling methods. The new method of Section 6.1 is based directly on the recurrence relations of Section 5, and always retains a positive-definite reduced Hessian. In contrast, Section 6.2 describes a method [22, 19] in which the reduced Hessian is allowed to be positive definite, singular or indefinite.

6.1. Updating an explicit positive-definite reduced Hessian. We now discuss an algorithm in which factorizations of A_* and of the (necessarily positive definite) matrix $Z_*^T H Z_*$ are used to solve the equations given in Section 5.1. We consider factorizations of A_* of the form

$$A_* Q_* = A_* \begin{pmatrix} Z_* & Y_* \end{pmatrix} = \begin{pmatrix} 0 & T \end{pmatrix}, \tag{6.1}$$

where T is a nonsingular $m_* \times m_*$ matrix, Q_* is an $n \times n$ nonsingular matrix, and Z_* and Y_* are the first $n - m_*$ and last m_* columns of Q.

Representing A_* by this factorization leads to simplification of the equations to be solved. In many implementations, Q_* is chosen so that T is triangular (see, e.g., Gill et al. [20]). In the reduced-gradient method, Q_* is defined so that T is the usual basis matrix B. The columns of Z_* form a basis for the null space of A_*. The columns of Y_* form a basis for the range space of A_*^T only if $Y_*^T Z_* = 0$.

Let $n_Z = n - m_*$. Let $\mathcal{Q}$ denote the (nonsingular) matrix

$$\mathcal{Q} = \begin{pmatrix} Q_* & \\ & I \end{pmatrix},$$

where I is the identity of dimension m_*. The n_Z-vector u_Z and the m_*-vector u_Y are defined by

$$u = Q_* \begin{pmatrix} u_Z \\ u_Y \end{pmatrix} = Z_* u_Z + Y_* u_Y. \tag{6.2}$$

Similarly,

$$y = Q_* \begin{pmatrix} y_Z \\ y_Y \end{pmatrix} \quad \text{and} \quad z = Q_* \begin{pmatrix} z_Z \\ z_Y \end{pmatrix}. \tag{6.3}$$

Multiplying (5.2) by $\mathcal{Q}^T$ and substituting from (6.1) and (6.2), we obtain

$$\begin{pmatrix} Q_*^T & \\ & I \end{pmatrix} \begin{pmatrix} H & A_*^T \\ A_* & \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} Q_*^T & \\ & I \end{pmatrix} \begin{pmatrix} g \\ 0 \end{pmatrix} =$$

$$\begin{pmatrix} Z_*^T H Z_* & Z_*^T H Y_* & 0 \\ Y_*^T H Z_* & Y_*^T H Y_* & T^T \\ 0 & T & \end{pmatrix} \begin{pmatrix} u_Z \\ u_Y \\ v \end{pmatrix} = \begin{pmatrix} Z^T g \\ Y^T g \\ 0 \end{pmatrix}. \tag{6.4}$$

Since T is nonsingular, the third equation of the partitioned system (6.4) implies that $u_Y = 0$, so that u and v are obtained by solving

$$Z_*^T H Z_* u_Z = Z_*^T g, \qquad T^T v = Y_*^T g + Y_*^T H Z_* u_Z, \tag{6.5}$$

and setting $u = Z_* u_Z$. The vectors z and t of (5.9) can similarly be found by solving

$$Z_*^T H Z_* z_Z = Z_*^T a, \qquad T^T t = Y_*^T a + Y_*^T H Z_* z_Z, \tag{6.6}$$

and setting $z = Z_* z_Z$.

We also need to compute the vectors y and w of (5.3) at a minimizer. Applying the same transformation as above and substituting from (6.3) gives the following equations to be solved:

$$Ty_Y = e_*, \quad Z_*^T H Z_* y_Z = -Z_*^T H Y_* y_Y, \quad T^T w = -Y_*^T H y, \tag{6.7}$$

where $y = Z_* y_Z + Y_* y_Y$.

By construction, the reduced Hessian $Z_*^T H Z_*$ is positive definite; let its Cholesky factorization be

$$Z_*^T H Z_* = R_*^T R_*, \tag{6.8}$$

where R_* is an upper-triangular matrix. An obvious strategy for a practical implementation is to retain the matrices T, Z_* and Y_* and the Cholesky factor R_*. As the iterations proceed, T, Z_* and Y_* can be updated to reflect changes in A_*, using Householder transformations or plane rotations if Q is orthogonal, and elementary transformations if Q is non-orthogonal; orthogonal transformations are needed in part of the update for R_* (see Gill *et al.* [20]).

For illustration, we sketch a particular updating technique in which T is chosen as *upper triangular*. In this discussion, barred quantities correspond to the "new" working set. When a constraint a added to the working set, a becomes the *first row* of A_*. To restore triangular form, we seek a matrix $\widetilde{Q}$ that annihilates the first $m_* - 1$ elements of $a^T Q_*$, i.e., such that

$$a^T Q_* \widetilde{Q} = \begin{pmatrix} a^T Z_* & a^T Y_* \end{pmatrix} \widetilde{Q} = \begin{pmatrix} 0 & \sigma & a^T Y_* \end{pmatrix}. \tag{6.9}$$

This result is achieved by choosing $\widetilde{Q}$ of the form

$$\widetilde{Q} = \begin{pmatrix} \widetilde{P} & 0 \\ 0 & I \end{pmatrix}, \tag{6.10}$$

where the $m_* \times m_*$ matrix $\widetilde{P}$ is composed of a sequence of orthogonal or elementary transformations. Substituting from (6.10) into (6.9), we have

$$\widetilde{P}^T Z_*^T a = \sigma e_{n_z}, \tag{6.11}$$

where e_{n_z} is the n_Z-th coordinate vector. The result is that

$$\bar{Q}_* = Q_* \widetilde{Q} = \begin{pmatrix} Z_* \widetilde{P} & Y_* \end{pmatrix} = \begin{pmatrix} \bar{Z}_* & \bar{Y}_* \end{pmatrix},$$

where

$$Z_* \widetilde{P} = \begin{pmatrix} \bar{Z}_* & \widetilde{y} \end{pmatrix}, \tag{6.12}$$

and $\bar{Y}_*$ is Y_* with a new first column (the transformed last column of Z_*).

When a constraint is deleted from A_*, the deleted row is moved to the first position by a sequence of cyclic *row* permutations, which need be applied only to T and Y_*. (The columns of Z_* are orthogonal to the rows of A_* in any order.) The first row of A_* may then be removed and the permuted triangle restored to proper form by transformations on the right without affecting the last $m_* - 1$ columns of Q_* or

T. The result is that $\bar{Y}_*$ is a row-permuted version of the last $m_* - 1$ columns of Y_*, and $\bar{Z}_*$ is given by

$$\bar{Z}_* = \begin{pmatrix} Z_* & \widetilde{z} \end{pmatrix}, \tag{6.13}$$

where $\widetilde{z}$ is a transformed version of the first column of Y_*.

This updating scheme leads to additional computational simplifications. For example, consider calculation of z and t from the first equation of (6.6) when a constraint is added to A_*. Multiplying by $\widetilde{P}^T$, substituting from (6.11), and letting $\widetilde{z} = \widetilde{P}^T z_Z$, $\widetilde{Z} = Z_* \widetilde{P}$, we have

$$\widetilde{Z}^T H \widetilde{Z} \widetilde{z} = \widetilde{Z}^T a = \sigma e_{n_z}. \tag{6.14}$$

The Cholesky factors $\widetilde{R}^T \widetilde{R}$ of $\widetilde{Z}^T H \widetilde{Z}$ will be available from the updating (see (6.12)), and the special form of the right-hand side of (6.14) means that the solve with the lower-triangular matrix $\widetilde{R}^T$ reduces to only a single division.

6.2. Updating a general reduced Hessian. In this section we briefly discuss the method of QPSOL [22], an inertia-controlling method based on maintaining an LDL^T factorization of the reduced Hessian

$$Z^T H Z = LDL^T, \tag{6.15}$$

where L is unit lower triangular and $D = \mathrm{diag}(d_j)$. When $Z^T H Z$ can be represented in the form (6.15), Sylvester's law of inertia (3.4) shows that $\mathrm{In}\,(Z^T H Z) = \mathrm{In}\,(D)$, and our inertia-controlling strategy thus ensures that D has at most one non-positive element. The following theorem states that, if the starting point is a minimizer, a null-space matrix Z exists such that only the *last* diagonal of D may be non-positive.

THEOREM 6.1. *Consider an inertia-controlling method in which the initial iterate x_0 is a minimizer. Then at every subsequent iterate there exist an upper-triangular matrix T, a unit lower-triangular matrix L, a diagonal matrix D and a null-space matrix Z with n_Z columns such that*

$$\begin{aligned} A \begin{pmatrix} Z & Y \end{pmatrix} &= \begin{pmatrix} 0 & T \end{pmatrix}, \\ Z^T H Z &= LDL^T, \\ Z^T g &= \sigma e_{n_z}, \end{aligned} \tag{6.16}$$

where $d_j > 0$ for $j = 1, \ldots, n_Z - 1$, and e_{n_z} is the n_Z-th coordinate vector.

Proof. An analogous result is proved by Gill and Murray [18] for a permuted form of the TQ factorization. □

We emphasize that the vector $Z^T g$ has the simple form (6.16) only when the TQ factorization of A is updated with elementary or plane rotation matrices applied in a certain order. In this sense, the method depends critically on the associated linear algebraic procedures.

The search direction p is always taken as a multiple of Zp_Z, where p_Z is the unique nonzero vector satisfying

$$L^T p_Z = e_{n_z}. \tag{6.17}$$

The special structures of D and the reduced gradient are crucial to the following theorem.

THEOREM 6.2. *Assume that the conditions of* Theorem 6.1 *hold, and let Z, L and D denote the matrices defined therein. Let p_Z be the solution of $L^T p_Z = e_{n_z}$, and let d_{n_z} denote the n_z-th diagonal element of D. Then the vector $p = Zp_Z$ is a multiple of q of* (5.4) *if $Z^T g \neq 0$ and $d_{n_z} \neq 0$, and is a multiple of y of* (5.3) *if either* (a) *$Z^T g \neq 0$ and $d_{n_z} = 0$, or* (b) *$Z^T g = 0$ and $d_{n_z} < 0$.*

Proof. In all cases, the definition (6.17) of p_Z and the structure of L and D imply that

$$LDL^T p_Z = d_{n_z} e_{n_z}. \tag{6.18}$$

First, assume that $Z^T g \neq 0$ and $d_{n_z} \neq 0$, so that $Z^T HZ$ is nonsingular and q is unique. Recall that $q = Zq_Z$, where $Z^T HZq_Z = Z^T g$. We know from Theorem 6.1 that $Z^T HZ = LDL^T$ and $Z^T g = \sigma e_{n_z}$, with $\sigma \neq 0$ by hypothesis. Relation (6.18) and the uniqueness of p and q thus imply that each is a multiple of the other, as required.

We now treat the second case, $Z^T g \neq 0$ and $d_{n_z} = 0$, so that $Z^T HZ$ is singular. The vector y of (5.3) can be written as $y = Zy_Z$, where y_Z is a *nonzero* vector satisfying $Z^T HZy_Z = 0$. (Recall that $Z^T HZ$ has exactly one zero eigenvalue.) Since $d_{n_z} = 0$, (6.18) gives

$$LDL^T p_Z = Z^T HZp_Z = 0,$$

as required.

Finally, assume that $Z^T g = 0$ and $d_{n_z} < 0$, which occurs when the reduced Hessian becomes indefinite immediately following deletion of a constraint with a zero multiplier. Let a_* be the normal of the deleted constraint with the zero multiplier. The vector y of (5.3) is given by $y = Zy_Z$, where y_Z satisfies

$$Z^T HZy_Z = -w_* Z^T a_* \quad \text{and} \quad a_*^T y = 1, \tag{6.19}$$

with $w_* > 0$. The nature of the updates to Z following a constraint deletion (see (6.13)) shows that the vector $Z^T a_*$ is given by

$$Z^T a_* = \xi e_{n_z}, \tag{6.20}$$

where $\xi = a_*^T \widetilde{z}$, with $\widetilde{z}$ the new column of Z created by the deletion of a_*. Because of the full rank of the working set, $\xi \neq 0$. Thus, y_Z satisfies

$$Z^T HZy_Z = -w_* \xi e_{n_z} \neq 0. \tag{6.21}$$

It follows from (6.18) that either p or $-p$ is a direction of negative curvature, since

$$p^T Hp = p_Z^T Z^T HZp_Z = d_{n_z} < 0.$$

If the sign of p_{n_z} (the last component of p_Z) is chosen so that

$$a_*^T p = a_*^T Zp_Z = \xi p_{n_z} > 0,$$

then examination of (6.18), (6.19) and (6.21) implies that p is a multiple of y, as required. □

Standard techniques are used to update the Cholesky factors (when the reduced Hessian is positive definite) and the LDL^T factors when the reduced Hessian is singular or indefinite. The test for "numerical" positive-definiteness is unavoidably scale-dependent, and involves a tolerance based on machine precision and the norm of the reduced Hessian. Because the usual bounds ensuring numerical stability do not apply when the reduced Hessian is indefinite, the last row and column of the reduced Hessian are *recomputed* if the last diagonal element of D is negative and "too large" following a constraint deletion (see the discussion in Gill and Murray [18]). Recomputation occurs when the last diagonal element is negative and its square exceeds a factor β ($\beta > 1$) times a measure of the norm of the reduced Hessian. (In the present version of QPSOL, $\beta = 10$.)

7. Conclusions and Topics for Further Research. This paper has explored in detail the nature of a family of methods for general quadratic programming. Our aims have been to describe the overall "feel" of an idealized active-set strategy (Section 2), to provide theoretical validation of the inertia-controlling strategy (Section 3), to formulate in a uniform notation the equations satisfied by the search direction (Section 4), and to discuss selected computational aspects of inertia-controlling methods (Section 5 and Section 6).

Many interesting topics remain to be explored, particularly in the efficient implementation of these methods. For example, the method of Section 6.1 is identical in motivation to Fletcher's original method [12], but has not been implemented in the form described, which avoids the need to update factors of a singular or indefinite symmetric matrix. Various methods for sparse quadratic programming could be devised based on the equations of Section 5.1, in addition to those already suggested by Gould [28] and Gill et al. [25].

As noted in Section 4.4, an open question remains concerning the crucial task of finding an initial working set in an efficient fashion consistent with the linear algebraic procedures of the main iterations.

Acknowledgement. The authors are grateful to Nick Gould and Anders Forsgren for many helpful discussions on quadratic programming. We also thank Dick Cottle for his bibliographical assistance.

REFERENCES

[1] T. Ando, *Generalized Schur complements*, Linear Algebra Appl., 27 (1979), pp. 173–186.
[2] M. J. Best, *Equivalence of some quadratic programming algorithms*, Math. Program., 30 (1984), pp. 71–87.
[3] J. R. Bunch and L. Kaufman, *A computational method for the indefinite quadratic programming problem*, Linear Algebra Appl., 34 (1980), pp. 341–370.
[4] S. Busovača, *Handling degeneracy in a nonlinear ℓ_1 algorithm*, Report CS-85-34, Department of Computer Science, University of Waterloo, Waterloo, Canada, 1985.
[5] D. Carlson, *What are Schur complements, anyway?*, Linear Algebra Appl., 74 (1986), pp. 257–275.
[6] D. Carlson, E. Haynsworth, and T. Markham, *A generalization of the Schur complement by means of the Moore-Penrose inverse*, SIAM J. Appl. Math., 26 (1974), pp. 169–175.
[7] L. B. Contesse, *Une caractérisation complète des minima locaux en programmation quadratique*, Numer. Math., 34 (1980), pp. 315–332.
[8] R. W. Cottle, *Manifestations of the Schur complement*, Linear Algebra Appl., 8 (1974), pp. 189–211.
[9] R. W. Cottle and A. Djang, *Algorithmic equivalence in quadratic programming, I: a least distance programming problem*, J. Optim. Theory Appl., 28 (1979), pp. 275–301.
[10] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.
[11] A. Dax, *The computation of descent directions at degenerate points*. Preprint, 1985.
[12] R. Fletcher, *A general quadratic programming algorithm*, J. Inst. Math. Applics., 7 (1971), pp. 76–91.

[13] ———, *Practical Methods of Optimization*, John Wiley and Sons, Chichester and New York, second ed., 1987.

[14] ———, *Recent developments in linear and quadratic programming*, in The State of the Art in Numerical Analysis, A. Iserles and M. J. D. Powell, eds., Oxford, 1987, Oxford University Press, pp. 213–243.

[15] ———, *Degeneracy in the presence of roundoff errors*, Linear Algebra Appl., 106 (1988), pp. 149–183.

[16] R. Fletcher and M. P. Jackson, *Minimization of a quadratic function of many variables subject only to upper and lower bounds*, J. Inst. Math. Applics., 14 (1974), pp. 159–174.

[17] A. Forsgren, P. E. Gill, and W. Murray, *On the identification of local minimizers in inertia-controlling methods for quadratic programming*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 730–746.

[18] P. E. Gill and W. Murray, *Numerically stable methods for quadratic programming*, Math. Program., 14 (1978), pp. 349–372.

[19] P. E. Gill, W. Murray, and M. A. Saunders, *User's guide for QPOPT 1.0: a Fortran package for quadratic programming*, Report SOL 95-4, Department of Operations Research, Stanford University, Stanford, CA, 1995.

[20] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Procedures for optimization problems with a mixture of bounds and general linear constraints*, ACM Trans. Math. Softw., 10 (1984), pp. 282–298.

[21] ———, *Sparse matrix methods in optimization*, SIAM J. Sci. and Statist. Comput., 5 (1984), pp. 562–589.

[22] ———, *User's guide for QPSOL (Version 3.2): a Fortran package for quadratic programming*, Report SOL 84-6, Department of Operations Research, Stanford University, Stanford, CA, 1984.

[23] ———, *User's guide for NPSOL (Version 4.0): a Fortran package for nonlinear programming*, Report SOL 86-2, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[24] ———, *A practical anti-cycling procedure for linearly constrained optimization*, Math. Program., 45 (1989), pp. 437–474.

[25] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *A Schur-complement method for sparse quadratic programming*, in Reliable Numerical Computation, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, 1990, pp. 113–138.

[26] D. Goldfarb and A. Idnani, *A numerically stable dual method for solving strictly convex quadratic programs*, Math. Programming, 27 (1983), pp. 1–33.

[27] N. I. M. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*, Math. Program., 32 (1985), pp. 90–99.

[28] ———, *On the convergence of a sequential penalty function method for constrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 107–128.

[29] N. I. M. Gould, *An algorithm for large-scale quadratic programming*, IMA J. Numer. Anal., 11 (1991), pp. 299–324.

[30] E. V. Haynsworth, *Determination of the inertia of a partitioned Hermitian matrix*, Linear Algebra Appl., 1 (1968), pp. 73–81.

[31] S. C. Hoyle, *A Single-Phase Method for Quadratic Programming*, PhD thesis, Report SOL 86-9, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[32] A. Majthay, *Optimality conditions for quadratic programming*, Math. Programming, 1 (1971), pp. 359–365.

[33] W. Murray, *An algorithm for finding a local minimum of an indefinite quadratic program*, Report NAC 1, National Physical Laboratory, England, 1971.

[34] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Program., 39 (1987), pp. 117–129.

[35] M. R. Osborne, *Finite Algorithms in Optimization and Data Analysis*, John Wiley and Sons, Chichester and New York, 1985.

[36] P. M. Pardalos and G. Schnitger, *Checking local optimality in constrained quadratic programming is NP-hard*, Operations Research Letters, 7 (1988), pp. 33–35.

[37] D. M. Ryan and M. R. Osborne, *On the solution of highly degenerate linear programs*, Math. Program., 41 (1986), pp. 385–392.

[38] J. Stoer, *A dual algorithm for solving degenerate linearly constrained linear least squares problems*, Report 47, Institut für Angewandte Mathematik und Statistik, Universität Würzburg, Germany, 1988.

[39] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.