

## ON MINIMIZING THE MAXIMUM EIGENVALUE OF A SYMMETRIC MATRIX\*

MICHAEL L. OVERTON†

**Abstract.** An important optimization problem that arises in control is to minimize  $\varphi(x)$ , the largest eigenvalue (in magnitude) of a symmetric matrix function of  $x$ . If the matrix function is affine,  $\varphi(x)$  is convex. However,  $\varphi(x)$  is not differentiable, since the eigenvalues are not differentiable at points where they coalesce. In this paper an algorithm that converges to the minimum of  $\varphi(x)$  at a quadratic rate is outlined. Second derivatives are not required to obtain quadratic convergence in cases where the solution is strongly unique. An important feature of the algorithm is the ability to split a multiple eigenvalue, if necessary, to obtain a descent direction. In these respects the new algorithm represents a significant improvement on the first-order methods of Polak and Wardi and of Doyle. The new method has much in common with the recent work of Fletcher on semidefinite constraints and Friedland, Nocedal, and Overton on inverse eigenvalue problems. Numerical examples are presented.

**Key words.** nonsmooth optimization, nondifferentiable optimization, convex programming, semidefinite constraints, minimizing maximum singular value

**AMS(MOS) subject classifications.** 65F99, 65K10, 90C25

**1. Introduction.** Many important optimization problems involve eigenvalue constraints. For example, in structural engineering we may wish to minimize the cost of some structure subject to constraints on its natural frequencies. A particularly common problem, which arises in control engineering, is

$$(1.1) \quad \min_{x \in \mathbb{R}^m} \varphi(x)$$

where

$$(1.2) \quad \varphi(x) = \max_{1 \leq i \leq n} |\lambda_i(A(x))|,$$

$A(x)$  is a real symmetric  $n \times n$  matrix-valued affine function of  $x$ , and

$$\{\lambda_i(A(x)), i = 1, \dots, n\}$$

are its eigenvalues. Since  $A(x)$  is an affine function, it may be written

$$A(x) = A_0 + \sum_{k=1}^m x_k A_k.$$

The function  $\varphi(x)$  is convex, since the largest eigenvalue of a matrix is a convex function of the matrix elements. An important special case is

$$(1.3) \quad A_k = e_k e_k^T$$

---

\* Received by the editors February 1, 1987; accepted for publication October 1, 1987. This work was supported in part by the National Science Foundation under grant DCR-85-02014. Some of the computer program development was performed at Stanford University, Stanford, California with support from the Office of Naval Research under contract ONR N00014-82-K-0335. This paper was presented at the SIAM Conference on Linear Algebra in Signals, Systems, and Control, which was held in Boston, Massachusetts on August 12-14, 1986.

† Courant Institute of Mathematical Sciences, New York University, New York, New York 10012. This work was completed while the author was on sabbatical leave at the Centre for Mathematical Analysis and Mathematical Sciences Research Institute, Australian National University, Canberra, Australia.

where  $e_k$  is the  $k$ th column of the identity matrix so that

$$(1.4) \quad A(x) = A_0 + \text{Diag}(x).$$

Note that the problem of minimizing the maximum singular value of a nonsymmetric matrix-valued affine function  $G(x)$  may be written in the form (1.1) since the eigenvalues of

$$\begin{bmatrix} 0 & G(x) \\ G(x)^T & 0 \end{bmatrix}$$

are (plus and minus) the singular values of  $G(x)$ . (Undoubtedly savings could be gained by treating the singular value problem more directly.)

The difficulty in minimizing  $\varphi(x)$  is that the function is not differentiable, since the eigenvalues are not differentiable quantities at points where they coalesce. Furthermore, we can usually expect the solution to be at a nondifferentiable point, since the minimization of  $\varphi(x)$  will generally drive several eigenvalues to the same minimum value.

In this paper we outline an algorithm that solves (1.1) with an asymptotic quadratic rate of convergence generically. Furthermore, second derivatives are not always required to obtain the quadratic convergence. In order to keep the paper fairly short we will not give proofs of convergence and we will omit some details of the algorithm, but the main ideas should be very clear. We believe this is the first time a quadratically convergent algorithm, or indeed any practical high-accuracy algorithm, has been described for minimizing  $\varphi(x)$ . An important feature of the algorithm is the ability to obtain a descent direction from any point that is not optimal, even if this requires splitting eigenvalues that are currently equal. (There are exceptions in degenerate cases.) This is also apparently new.

In these respects the algorithm given here represents a significant improvement on the first-order methods for the same problem described by Polak and Wardi (1982) and Doyle (1982). The present paper is heavily influenced by two works, Fletcher (1985) and Friedland, Nocedal, and Overton (1987), to which full acknowledgment is given. Personal communication with Doyle was also very helpful. Another important early reference is Cullum, Donath, and Wolfe (1975), who give a first-order method for a related problem. Undoubtedly a variant of the algorithm given here could be derived for that problem. Finally, we should not overlook the related structural engineering literature (see Olhoff and Taylor (1983, p. 1146) for a useful survey).

**2. Connections with the work of Fletcher and Friedland, Nocedal, and Overton.** The problem (1.1) may be rewritten as the nondifferentiable constrained optimization problem

$$(2.1) \quad \min_{\omega \in \mathbb{R}, x \in \mathbb{R}^m} \omega$$

$$(2.2) \quad \text{s.t.} \quad -\omega \leq \lambda_i(A(x)) \leq \omega, \quad i = 1, \dots, n,$$

or equivalently

$$(2.3) \quad \min_{\omega \in \mathbb{R}, x \in \mathbb{R}^m} \omega$$

$$(2.4) \quad \text{s.t.} \quad \omega I - A(x) \geq 0,$$

$$(2.5) \quad \omega I + A(x) \geq 0$$

where “ $\geq$ ” in (2.4), (2.5) indicates a matrix positive semidefinite constraint. The second formulation immediately suggests that the work of Fletcher (1985) is applicable. Fletcher gives a quadratically convergent algorithm to solve

$$(2.6) \quad \max \sum_{i=1}^m x_k$$

$$(2.7) \quad \text{s.t. } A_0 - \text{Diag}(x) \geq 0, \quad x \geq 0$$

and many of the components of his algorithm are therefore applicable to solving (2.3)–(2.5). However, the algorithm is not directly applicable and there are several reasons why it is possible to improve on Fletcher’s method in this case. One reason is that Fletcher’s method solves a sequence of subproblems, each defined by a guess of the nullity of  $A_0 + \text{Diag}(x)$ , until the correct nullity is identified. Such a strategy cannot easily be extended to the case of two (or more) semidefinite constraints. One goal of our algorithm is to be able to adjust multiplicity estimates while always obtaining a reduction of  $\varphi(x)$  at each iteration. We are able to do this by computing an eigenvalue-eigenvector factorization of  $A(x)$  at each iteration. By contrast, Fletcher’s method uses a block Choleski factorization of  $A_0 + \text{Diag}(x)$ , together with an exact penalty function to impose (2.7).

Also, because of the special form of (2.6), (2.7), Fletcher’s method does not require a technique for splitting eigenvalues. In other words, given a matrix  $A_0 + \text{Diag}(x)$ , satisfying (2.7), with nullity  $t$ , it cannot be advantageous, in the sense of increasing (2.6), to reduce the multiplicity  $t$ . On the other hand it may be necessary to split a multiple eigenvalue in our case, and the ability to recognize this situation and obtain an appropriate descent direction is an important part of our algorithm.

Because we use an eigenvalue factorization of the matrix  $A(x)$  at each iterate  $x$ , our method has much in common with the methods described by Friedland, Nocedal, and Overton (1987). In the latter paper several quadratically convergent methods are given to solve

$$(2.8) \quad \lambda_i(A(x)) = \omega, \quad i = 1, \dots, t,$$

$$(2.9) \quad \lambda_i(A(x)) = \mu_i, \quad i = t + 1, \dots, \hat{t}$$

where  $(\omega, \{\mu_i\})$  are given distinct values and  $t, \hat{t}$  (and  $m$ , the number of variables) are appropriately chosen. One of the contributions of that paper was to explain that the condition (2.8), although apparently only  $t$  conditions, actually generically imposes  $t(t + 1)/2$  linearly independent constraints on the parameter space, and that effective numerical methods must be based on this consideration. The present paper may be viewed as generalizing the methods of Friedland, Nocedal, and Overton to solve

$$(2.10) \quad \min_{\omega \in \mathbb{R}, x \in \mathbb{R}^m} \omega$$

$$(2.11) \quad \text{s.t. } \lambda_i(A(x)) = \omega, \quad i = 1, \dots, t,$$

$$(2.12) \quad \lambda_i(A(x)) = -\omega, \quad i = n - s + 1, \dots, n$$

where, as a product of the minimization process, it is established that  $\omega = \max(\lambda_1, -\lambda_n)$  with

$$(2.13) \quad \omega = \lambda_1 = \dots = \lambda_t > \lambda_{t+1} \geq \dots \geq \lambda_{n-s} > \lambda_{n-s+1} = \dots = \lambda_n = -\omega.$$

We shall subsequently refer to  $t$  and  $s$  as the upper and lower (eigenvalue) multiplicities of  $A(x)$ . Note that it is possible that either  $t$  or  $s$  is zero. The following notation will be useful subsequently: let  $\{q_1(x), \dots, q_n(x)\}$  be any orthonormal set of eigenvectors of  $A(x)$  corresponding to  $\{\lambda_i\}$ , and let  $Q_1 = [q_1, \dots, q_t]$ ,  $Q_2 = [q_{n-s+1}, \dots, q_n]$ .

**3. Optimality conditions.** As Fletcher points out, it is convenient to initially consider the variable space to be the set of all  $n \times n$  symmetric matrices  $\{A\}$  and to consider the positive semidefinite cone

$$(3.1) \quad K = \{A | A \geq 0\}.$$

Define an inner product on the set of symmetric matrices by

$$(3.2) \quad A:B = \text{tr } AB = \sum_{i,j} a_{ij}b_{ij}.$$

The normal cone (Rockafellar (1970)) is defined by

$$(3.3) \quad \partial K(A') = \{B | A':B = \sup_{A \in K} A:B\}.$$

Fletcher shows that a very useful expression for  $\partial K$  is

$$(3.4) \quad \partial K(A') = \{B | B = -ZUZ^T, U = U^T, U \geq 0\}$$

where the columns of  $Z$  span the null space of  $A'$ .

Now consider the restricted variable spaces

$$(3.5) \quad \hat{K}_1 = \{(\omega, x) | \omega I - A(x) \geq 0; \omega \in \mathbb{R}; x \in \mathbb{R}^m\},$$

$$(3.6) \quad \hat{K}_2 = \{(\omega, x) | \omega I + A(x) \geq 0; \omega \in \mathbb{R}; x \in \mathbb{R}^m\}.$$

By definition,

$$(3.7) \quad \partial \hat{K}_1(\omega', x') = \{(\delta, d) | (\omega', x')^T(\delta, d) = \sup_{(\omega, x) \in \hat{K}_1} (\omega, x)^T(\delta, d)\}.$$

**THEOREM 3.1.**

$$(3.8) \quad \partial \hat{K}_1(\omega', x') = \{(\delta, d) | \delta = B:I; d_k = -B:A_k, k = 1, \dots, m, \\ B \in \partial K(\omega'I - A(x'))\}.$$

*Proof.* The proof is omitted because it is almost identical to the proof of Fletcher's Theorem 4.1. Fletcher's proof essentially covers the special case (1.3). One important point worth mentioning is that Fletcher's construction of a feasible arc may require an augmenting term  $\alpha \varepsilon^2 I$  in the arc parameterization; in our case this may be absorbed by the  $\omega I$  term in  $\omega I - A(x)$ .  $\square$

We can now state the optimality condition for  $x$  to solve (1.1).

**THEOREM 3.2.** *A necessary and sufficient condition for  $x$  to solve (1.1) is that there exist matrices  $U$  and  $V$  of dimension  $t \times t$  and  $s \times s$ , respectively, with  $U = U^T \geq 0$ ,  $V = V^T \geq 0$ , such that*

$$(3.9) \quad \text{tr } U + \text{tr } V = 1,$$

$$(3.10) \quad (Q_1^T A_k Q_1):U - (Q_2^T A_k Q_2):V = 0, \quad k = 1, \dots, m.$$

Here  $t, s, Q_1, Q_2$  are defined by (2.13) and the following remarks.

*Proof.* Because of the equivalence of (1.1) with the convex problem (2.3)–(2.5), the necessary and sufficient condition for optimality is

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + g_1 + g_2 = 0$$

where  $g_1 \in \partial\hat{K}_1$  and  $g_2 \in \partial\hat{K}_2$  (Rockafellar (1981, Chap. 5)). By Theorem 3.1 we therefore require

$$1 + \text{tr } B_1 + \text{tr } B_2 = 0,$$

$$-B_1 : A_k + B_2 : A_k = 0, \quad k = 1, \dots, m$$

where  $B_1 \in \partial K(\omega I - A(x))$ ,  $B_2 \in \partial K(\omega I + A(x))$ ,  $\omega = \max(\lambda_1(A(x)), -\lambda_n(A(x)))$ . By (3.4) we have

$$B_1 = -Q_1 U Q_1^T, \quad B_2 = -Q_2 V Q_2^T$$

for some  $U \geq 0$  and  $V \geq 0$ , since  $Q_1$  is a basis for the null space of  $\omega I - A(x)$  and  $Q_2$  for  $\omega I + A(x)$ . Now as Fletcher points out,  $U : (Z^T A Z) = A : (Z U Z^T)$  for any  $A \in \mathbb{R}^{n \times n}$ ,  $U \in \mathbb{R}^{t \times t}$ , and  $Z \in \mathbb{R}^{n \times t}$ . (A proof is as follows:  $U : Z^T A Z = \text{tr } U Z^T A Z = \text{tr } Z (U Z^T A) = (Z U Z^T) : A$ , where the middle equality holds because  $\text{tr } Z P = \text{tr } P Z$ , where  $Z \in \mathbb{R}^{n \times t}$ ,  $P \in \mathbb{R}^{t \times n}$ .) The theorem is therefore proved.  $\square$

The matrices  $U$  and  $V$  in (3.9) and (3.10) play the role of Lagrange multipliers, as will become clear in the next section. Because the optimality conditions  $U \geq 0$ ,  $V \geq 0$  are conditions on the matrices as a whole, rather than componentwise conditions, we call  $U$  and  $V$  Lagrange matrices (cf. ‘‘Lagrange vectors’’ in Overton (1983)).

**4. An algorithm based on successive quadratic programming.** As explained by Friedland, Nocedal, and Overton, a quadratically convergent method for solving the nondifferentiable system (2.8), (2.9) may be obtained by applying a variant of Newton’s method to the nonlinear but essentially differentiable system

$$(4.1) \quad Q_1(x)^T A(x) Q_1(x) = \omega I_t,$$

$$(4.2) \quad q_i(x)^T A(x) q_i(x) = \mu_i, \quad i = t + 1, \dots, \hat{t}$$

where the columns of  $Q_1(x)$  are an orthonormal set of eigenvectors of  $A(x)$  corresponding to  $\omega$ . Here  $I_t$  denotes the identity matrix of order  $t$ . Let  $x^*$  satisfy (4.1), (4.2). Note that (4.1) is independent of the choice of basis for  $Q_1(x^*)$ . Also note that for points in a neighbourhood of  $x^*$ ,  $A(x)$  will generally have distinct eigenvalues (with small separation) and hence  $Q_1(x)$ , the matrix of eigenvectors corresponding to the multiple eigenvalue at  $x^*$ , will be a well-defined but ill-conditioned function of  $x$  which does not converge as  $x \rightarrow x^*$ . This does not cause any difficulties for the Newton method (see Friedland, Nocedal, and Overton (1987) for details). In order to obtain quadratic convergence we need the number of equations,  $t(t + 1)/2 + (\hat{t} - t)$ , to equal the number of variables (together with a nonsingularity condition). When we differentiate (4.1), (4.2), we find that the appropriate system of equations to solve at each step of the Newton method is

$$Q_1(x)^T A(x + d) Q_1(x) = \omega I_t,$$

$$q_i(x)^T A(x + d) q_i(x) = \mu_i, \quad i = t + 1, \dots, \hat{t}$$

where  $x$  is the current iterate and  $x + d$  becomes the new iterate. (Although this may look counterintuitive, note that the left-hand side of (4.2) is simply  $\lambda_i(x)$ , and hence the latter equation is consistent with the well-known fact that the derivative of  $\lambda_i(x)$  (with respect to  $x_k$ ) is  $q_i(x)^T A_k q_i(x)$ . Again, see Friedland, Nocedal, and Overton (1987) for details.) Since  $A(x + d)$  is affine, these equations form a linear system in  $d$ . Once  $x + d$  is obtained, an eigenvalue-eigenvector factorization of  $A$  at the new point is required to be able to start the next iteration.

Now consider generalizing this method to solve (2.10)–(2.12), where we assume for the moment that  $t$  and  $s$  are known. We see that the Newton method should be applied to the nonlinear problem

$$(4.3) \quad \min_{\omega \in \mathbb{R}, x \in \mathbb{R}^m} \omega$$

$$(4.4) \quad \text{s.t.} \quad \omega I_t - Q_1(x)^T A(x) Q_1(x) = 0,$$

$$(4.5) \quad \omega I_s + Q_2(x)^T A(x) Q_2(x) = 0.$$

The appropriate subproblem to solve at each step of the Newton method is the quadratic program (QP)

$$(4.6) \quad \min_{\omega \in \mathbb{R}, d \in \mathbb{R}^m} \omega + \frac{1}{2} d^T W d$$

$$(4.7) \quad \text{s.t.} \quad \omega I_t - Q_1(x)^T A(x + d) Q_1(x) = 0,$$

$$(4.8) \quad \omega I_s + Q_2(x)^T A(x + d) Q_2(x) = 0$$

where  $W$  is a matrix to be specified shortly.

Now define a Lagrangian function for (4.3)–(4.5) by

$$(4.9) \quad L(\omega, x, U, V) = \omega - U: (\omega I_t - Q_1(x)^T A(x) Q_1(x)) - V: (\omega I_s + Q_2(x)^T A(x) Q_2(x))$$

where  $U = U^T$ ,  $V = V^T$ . Since (4.7)–(4.8) represent a linearization of (4.4)–(4.5), we see that the first-order necessary condition for  $x$  to solve (4.3)–(4.5), namely  $\nabla_{\omega, x} L = 0$ , is that there exist symmetric matrices  $U$  and  $V$  such that (3.9)–(3.10) hold—the same optimality condition given in the previous section. (Similarly, if a sequence of QPs (4.6)–(4.8) has been solved, converging to a solution of (4.3)–(4.5) and hence with  $d$  converging to zero, the optimality condition of the limiting QP is that there exist  $U$  and  $V$  such that (3.9)–(3.10) hold.) The equivalence with the optimality conditions (3.9)–(3.10) is very important, since it means that the Lagrange matrices required to check the optimality conditions (3.9), (3.10) may be obtained by solving (4.3)–(4.5), or more specifically, by solving a sequence of QPs (4.6)–(4.8). This observation is the same as the one emphasized by Fletcher and is the essential justification for an algorithm based on successive quadratic programming (SQP). The key point is that (4.3)–(4.5) is much more tractable than the original problem. A related point to note is that  $U$  and  $V$  are not required to be positive semidefinite for an optimal solution to (4.3)–(4.5), since the constraints are equalities. If  $U$  or  $V$  is indefinite, this is an indication that  $t$  or  $s$  is too large and that it is necessary to split a multiple eigenvalue, as will be explained in § 5.

The number of constraints in (4.4), (4.5) is

$$(4.10) \quad \frac{t(t+1)}{2} + \frac{s(s+1)}{2}.$$

If this quantity is equal to  $m + 1$  (the number of variables in (4.3)), then, generically, the constraints themselves are enough to define a unique solution to (4.3)–(4.5) and the SQP method will have local quadratic convergence regardless of the value of  $W$ . In this case the solution of (1.1) is “strongly unique.” If (4.10) is greater than  $m + 1$ , then, except in degenerate cases, (4.3)–(4.5) will be infeasible. In general we expect (4.10) to be less than or equal to  $m + 1$ , but we cannot expect equality—for example, if  $m = 4$ , equality is not possible. If (4.10) is less than  $m + 1$  the proper choice of the matrix  $W$  is necessary for the SQP method to converge quadratically. It is clear that  $W$  should be set to the

Hessian, with respect to  $x$ , of the Lagrangian function (4.9). It can be shown that this matrix is given by

$$(4.11) \quad W_{jk} = U:G_1^{j,k} - V:G_2^{j,k}$$

where

$$(4.12) \quad G_l^{j,k} = 2Q_l(x)^T A_k \bar{Q}_l(x) (\omega J_l - \bar{\Lambda}_l(x))^{-1} \bar{Q}_l(x)^T A_j Q_l(x), \quad l = 1, 2,$$

and where the columns of  $\bar{Q}_l(x)$  consist of all eigenvectors in  $\{q_1, \dots, q_n\}$  except those in  $Q_l(x)$  ( $l = 1, 2$ ),  $\bar{\Lambda}_l(x)$  is a diagonal matrix whose entries consist of all eigenvalues in  $\{\lambda_1, \dots, \lambda_n\}$  except those corresponding to  $Q_l(x)$ , and  $J_1 = I_{n-t}$ ,  $J_2 = -I_{n-s}$ .

Some caution is required in the choice of  $U$  and  $V$  in (4.11). Since the values of  $U$  and  $V$  at the minimum of (4.3)–(4.5) are not known, it is necessary to use Lagrange matrix estimates. The obvious choice is to use the values obtained from the previous QP. Unfortunately these are useless, because the eigenvector bases  $Q_1(x)$  and  $Q_2(x)$  at the current point will generally have no relation to those at the previous point (although the range spaces of  $Q_1(x)$  and  $Q_2(x)$  will converge as  $x$  converges to a minimizing point). Therefore, after  $Q_1(x)$ ,  $Q_2(x)$  have been computed, but before solving the QP, it is necessary to obtain first-order Lagrange matrix estimates by minimizing the 2-norm of the residual of (3.9), (3.10). This does not require a significant amount of extra work since the QR factorization of the relevant coefficient matrix is needed anyway to solve the QP. (See Murray and Overton (1980) for some comments on Lagrange multiplier estimates for minimax problems and Nocedal and Overton (1985) for comments on first- and second-order Lagrange multiplier estimates.) Alternatively, we could use the Cayley transform method (“Method III”) of Friedland, Nocedal, and Overton, which updates estimates of the eigenvectors without recomputing them in such a way that even the eigenvector estimates corresponding to multiple eigenvalues converge. This technique does not impede quadratic convergence. It would be essential if we wanted to use a quasi-Newton method to approximate the matrix  $W$  without computing (4.11), which might be necessary for large problems. Note again, however, that  $W$  is not needed at all if (4.10) equals  $m + 1$ , which may quite often be the case.

We now turn to the important question of how the upper and lower multiplicities  $t$  and  $s$  are to be determined. These can be effectively obtained dynamically. Suppose that  $t = 1$ ,  $s = 0$  initially. If the QP (4.6)–(4.8) were now to be solved, the solution would very likely reduce the initially largest eigenvalue far below the others. It is therefore sensible to incorporate into the QP inequality constraints on the other eigenvalues, namely

$$(4.13) \quad -\omega \leq q_i^T(x)A(x+d)q_i(x) \leq \omega, \quad t + 1 \leq i \leq n - s.$$

We may now obtain updated estimates of  $t$  and  $s$  by seeing which constraints are active at the solution of the QP (4.6)–(4.8), (4.13). A reasonable strategy is to increase  $t$  by the number of constraints which are at their upper bound and to increase  $s$  by the number at their lower bound. However, some caution should be used, since if  $t$  and  $s$  become too large, (4.3)–(4.5) will become infeasible. We therefore also keep more conservative estimates  $\bar{t}$  and  $\bar{s}$  which are defined at the beginning of each iteration by

$$(4.14) \quad \omega - \lambda_i(x) \leq \text{TOL}, \quad i = 1, \dots, \bar{t},$$

$$(4.15) \quad \omega + \lambda_i(x) \leq \text{TOL}, \quad i = n - \bar{s} + 1, \dots, n$$

assuming that  $\bar{t}(\bar{t} + 1)/2 + \bar{s}(\bar{s} + 1)/2 \leq m + 1$ , where  $\omega = \varphi(x)$  and TOL is a reasonably small number, e.g.,  $10^{-2}$ . If necessary  $t$  and  $s$  are reset to these more conservative values, as will be explained shortly. However, if  $t$  and  $s$  are always set to  $\bar{t}$  and  $\bar{s}$  instead of making use of the active constraint information from the solution of the previous QP, the al-

gorithm, though reliable, converges much more slowly, since quadratic convergence cannot take place until the eigenvalue separation is reduced to TOL (unless the solution has distinct eigenvalues). (A possible alternative approach to accelerating the selection of the correct multiplicity estimates would be to use a special line search as is done in Overton (1983).)

At each iteration we insist that a reduction in  $\omega$  is obtained. Even if  $t$  and  $s$  have the correct values defined by (2.13), there is no guarantee that the solution  $d$  of (4.6)–(4.8), (4.13) will give  $\varphi(x + d) < \varphi(x)$ . Following Fletcher, we therefore use a “trust region” strategy, incorporating into the QP bound constraints

$$(4.16) \quad |d_k| \leq \rho, \quad k = 1, \dots, m$$

where  $\rho$  is dynamically adjusted. It is clear that if  $\rho$  and TOL are sufficiently small, then the solution  $d$  of the QP (4.6)–(4.8), (4.13), (4.16), with  $t = \bar{t}$ ,  $s = \bar{s}$ , will give  $\varphi(x + d) < \varphi(x)$  unless  $d = 0$ .

If TOL = 0,  $t = \bar{t}$ ,  $s = \bar{s}$  and the solution  $d$  of the QP is zero, the point  $x$  is a minimum of (4.3)–(4.5). It therefore also solves (1.1) if the Lagrange matrices  $U$  and  $V$  are positive semidefinite. If  $U$  or  $V$  is indefinite then it is both necessary and feasible to split a multiple eigenvalue to make further progress, as will be explained in § 5.

We conclude this section with a summary of the algorithm. It requires initial values for TOL and  $\rho$  and a convergence tolerance  $\epsilon$ .

ALGORITHM.

0. Given  $x$ , evaluate  $\{\lambda_i(x)\}$ ,  $\{q_i(x)\}$ . Define  $\bar{t}$ ,  $\bar{s}$  by (4.14), (4.15). Set  $t = \bar{t}$ ,  $s = \bar{s}$ .
1. Solve the QP (4.6)–(4.8), (4.13), (4.16), using first-order Lagrange matrix estimates to define  $W$ . If the QP is infeasible, go to Step 2.2. If  $\|d\| \leq \epsilon$ , go to Step 3.
2. Evaluate  $\{\lambda_i(x + d)\}$ . If  $\varphi(x + d) < \varphi(x)$ , then
  - 2.1 Increase  $t$  and  $s$ , respectively, by the number of upper and lower bounds which are active in (4.13), provided the new values give (4.10) less than or equal to  $m + 1$ . Set  $x$  to  $x + d$ , evaluate  $\{q_i(x)\}$ , and define  $\bar{t}$ ,  $\bar{s}$  by (4.14), (4.15). Double  $\rho$ , and go to Step 1.
- else
  - 2.2 Reset  $t$ ,  $s$  to  $\bar{t}$ ,  $\bar{s}$ . Divide  $\rho$  by two and go to Step 1.
3. If  $U \geq 0$  and  $V \geq 0$  then
  - 3.1 STOP –  $x$  is optimal.
- else
  - 3.2 Split a multiple eigenvalue and obtain reduction as described in the next section. Adjust  $\bar{t}$ ,  $\bar{s}$ ,  $t$ ,  $s$  accordingly and go to Step 1.

This algorithm has worked well in practice (see the results in § 6). Clearly it can be defeated; in particular, if TOL is not small enough, the QP may be infeasible, and at present there is no facility for reducing TOL. However, it seems likely that it will form the basis of a more elaborate algorithm for which global convergence can be guaranteed. Because  $\varphi(x)$  is convex, obtaining a globally convergent algorithm is not difficult; what is wanted is a globally convergent algorithm for which final quadratic convergence is guaranteed (given nonsingularity assumptions).

**5. Splitting multiple eigenvalues.** Consider a simple example. Let  $m = n = 2$ , with

$$(5.1) \quad A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & \kappa \\ \kappa & 4 \end{bmatrix}$$



for some value  $\kappa$ . Since  $A(x) = A_0 + x_1A_1 + x_2A_2$ , the only point where  $A(x)$  has multiple eigenvalues is  $x = (0, 0)^T$ , which is therefore a solution of (4.3)–(4.5) with  $t = 2, s = 0$ . If  $\kappa$  is large enough, clearly  $x = (0, 0)^T$  is a minimum of  $\varphi(x)$ , since  $x_1A_1 + x_2A_2$  is indefinite for any nonzero  $x$ . On the other hand, if  $\kappa$  is small enough,  $A_2$  is positive definite, and  $d = (0, -1)^T$  is a descent direction from  $x = (0, 0)^T$ . It is therefore essential to be able to distinguish between these situations and to find a descent direction if one exists. It appears that an inability to do this has been one of the major deficiencies of algorithms previously developed for (1.1) (Doyle (1986)).

In order to check optimality, we introduce the Lagrange matrix  $U$  ( $V$  is empty since  $s = 0$ ). The system (3.9), (3.10) is

$$\begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & -4 & -\kappa \end{bmatrix} \begin{bmatrix} U_{11} \\ U_{22} \\ 2U_{12} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

where we arbitrarily choose  $Q_1 = I$ . The solution is

$$(5.2) \quad U = \begin{bmatrix} \frac{1}{2} & \frac{-5}{4\kappa} \\ \frac{-5}{4\kappa} & \frac{1}{2} \end{bmatrix}.$$

The optimality condition is  $U \geq 0$ , i.e.,

$$|\kappa| \geq \frac{5}{2}.$$

We now show how to obtain a descent direction if  $|\kappa| < \frac{5}{2}$ . The solution is to solve

$$\delta I - \sum_{k=1}^2 d_k A_k = -\mu uu^T$$

where  $\mu$  is the negative eigenvalue of  $U$  and  $u$  is the corresponding eigenvector. This gives, in the case of  $\kappa = 2.25$ ,

$$\begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & -4 \\ 0 & 0 & -\kappa \end{bmatrix} \begin{bmatrix} \delta \\ d_1 \\ d_2 \end{bmatrix} = 2.78 \times 10^{-2} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

i.e.,  $\delta = -3.09 \times 10^{-3}, d = (-1.85 \times 10^{-2}, -1.23 \times 10^{-2})^T$ . Now  $\lambda(A(x + d)) = (0.941, 0.997)^T$  so that  $\varphi(x + d) < \varphi(x)$  as required. Note that  $d = (0, -1)^T$  is not a descent direction from  $x = 0$  in this case.

More generally, we have the following.

**THEOREM 5.1.** *Let  $t$  and  $s$  be defined by (2.13). Assume  $x$  is a minimum of (4.3)–(4.5), so that (3.9), (3.10) hold for some symmetric matrices  $U \in \mathbb{R}^{t \times t}$  and  $V \in \mathbb{R}^{s \times s}$ . Suppose that  $U$  is indefinite with a negative eigenvalue  $\mu$  and corresponding eigenvector  $u$ . Then if  $(\delta, d)$  solves*

$$(5.3) \quad \delta I_t - \sum_{k=1}^m d_k Q_1^T A_k Q_1 = -\mu uu^T,$$

$$(5.4) \quad \delta I_s + \sum_{k=1}^m d_k Q_2^T A_k Q_2 = 0,$$

we have that  $d$  is a descent direction for  $\varphi(x)$ . Furthermore, to first order the multiplicity  $t$  is reduced by exactly one along  $d$ , and the new set of eigenvectors for  $\lambda_i = \omega$  can be taken, to first order, as

$$\tilde{Q}_1 = Q_1 \bar{U}$$

where the columns of  $\bar{U}$  are the eigenvectors of  $U$ , excluding  $u$ .

*Remark.* Equations (5.3)–(5.4) are generically solvable if (4.10) is less than or equal to  $m + 1$ . Other cases are degenerate situations for which obtaining a descent direction is more difficult.

*Proof.* Taking an inner product of  $U$  with (5.3) and  $V$  with (5.4) and adding them together we obtain

$$\delta(\text{tr } U + \text{tr } V) + \sum_{k=1}^m d_k(-U:Q_1^T A_k Q_1 + V:Q_2^T A_k Q_2) = -\mu^2.$$

It therefore follows from (3.9), (3.10) that

$$(5.5) \quad \delta = -\mu^2.$$

Furthermore, for the same reason that (4.7) is a valid linearization of (4.4), (5.3), and (5.4) show that the constraints (2.4), (2.5) hold to first order along the direction  $x + \alpha d$ ,  $\alpha \geq 0$  (since the right-hand sides of (5.3), (5.4) are positive semidefinite). It follows from (5.5) that  $d$  is a descent direction. Finally, the last statement is justified by multiplying (5.3) by  $(u, \bar{U})^T$  on the left and  $(u, \bar{U})$  on the right, obtaining

$$\delta I_t - \sum_{k=1}^m d_k(u, \bar{U})^T Q_1^T A_k Q_1 (u, \bar{U}) = \begin{bmatrix} -\mu & \\ & 0 \end{bmatrix}. \quad \square$$

In other words, all eigenvalues but one are reduced by  $\mu^2$  (to first order) while the other eigenvalue is reduced by  $\mu^2 - \mu$ .

More generally still, if  $U$  has more than one negative eigenvalue (or  $U$  and  $V$  both have negative eigenvalues), we can reduce  $t$  by more than one (or reduce both  $t$  and  $s$ ) by replacing the right-hand side of (5.3) (and (5.4)) by a sum of outer products corresponding to the negative eigenvalues. This has an obvious analogy in nonlinear programming, where if several Lagrange multipliers are negative at a stationary point we can move off just a single constraint (as does the simplex method for linear programming) or move off several constraints at once. Also, in nonlinear programming we may move off a constraint before minimizing on the corresponding manifold if the appropriate Lagrange multiplier estimate is negative. Similarly, we should be able to use Lagrange matrix estimates to avoid minimizing on the manifold defined by (4.4), (4.5).

**6. Numerical examples.** The algorithm has been implemented in Fortran and run on a Pyramid Unix system at Australian National University. Double precision arithmetic (about 15 decimal digits of accuracy) was used. The eigensystems of  $A(x)$  were computed using EISPACK (Smith et al. (1967)). The QPs were solved using the Stanford package QPSOL (Gill et al. (1984)).

We give three examples that illustrate the effectiveness of the method. The parameters  $\epsilon$  and TOL were given the values  $10^{-7}$  and  $10^{-2}$ , respectively, and the initial trust region radius  $\rho$  was set to 1. The tables shown below have the following meaning. There is one row in the table for each time a reduction in  $\varphi(x + d)$  is obtained, i.e., Step 2.1 is executed. The values  $\bar{t}$ ,  $\bar{s}$ ,  $t$ , and  $s$  are those holding at the beginning of the iteration, i.e., following the previous execution of Step 0 or 2.1. The quantity #QPs is the number of QPs that

had to be solved before obtaining a reduction, i.e., the number of times Step 1 was executed. Step 3.2 was not executed in any of these examples.

*Example 1.* This is defined by (5.1) with  $\kappa = 3$ .

Initial  $x = (1.0, 2.0)^T$  with  $\varphi(x) = 12.32$ .

Iteration	$\bar{t}$	$\bar{s}$	$t$	$s$	#QPs	$\varphi(x + d)$
1	1	0	1	0	1	6.541381
2	1	0	1	0	1	4.817767
3	1	0	2	0	1	1.000000

Final  $x = (0.0, 2.0 \times 10^{-15})^T$  with  $\lambda(x) = (1.0, 1.0)^T$  and

$$U = \begin{bmatrix} 0.8716 & -0.1884 \\ -0.1884 & 0.1284 \end{bmatrix}.$$

*Comments.* Once the correct multiplicities are identified this particular problem is solved in one step. The reason that  $U$  is different from (5.2) is that EISPACK chose a basis  $Q_1 \neq I$ . Of course this does not affect the optimality condition.

*Example 2.*  $n = 3, m = 3$ ,

$$A_0 = \begin{bmatrix} 0 & 1.0 & 1.1 \\ 1.0 & 0 & 1.2 \\ 1.1 & 1.2 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix}.$$

Initial  $x = (1.0, 0.9, 0.8)^T$  with  $\varphi(x) = 7.605$ .

Iteration	$\bar{t}$	$\bar{s}$	$t$	$s$	#QPs	$\varphi(x + d)$
1	1	0	1	0	1	1.616283
2	1	0	1	0	1	1.464941
3	0	1	1	2	1	1.145090
4	1	1	1	2	1	1.102385
5	1	2	1	2	1	1.101521
6	1	2	1	2	1	1.101520

Final  $x = (-0.1163679, -0.2497934, -1.845989)^T$  with

$$\lambda(x) = (1.101520, -1.101520, -1.101520)^T$$

and

$$U = [6.95 \times 10^{-4}], \quad V = \begin{bmatrix} 0.4861 & 0.0229 \\ 0.0229 & 0.5132 \end{bmatrix}.$$

*Comments.* Note that  $U$  is only barely positive definite, so that a small perturbation to the problem would give an optimal point with  $\lambda_1 < \omega$ . As in Example 1, (4.10) equals  $m + 1$  at the solution, so  $W$  is not needed for quadratic convergence, although it may help to identify  $t$  and  $s$  during the early iterations. Note also that following the first iteration where the correct multiplicities were used to define the QP, the solution is correct to two figures.

Example 3.  $n = 10, m = 10, A_k = e_k e_k^T, k = 1, \dots, 10,$  and

$$A_0 = \begin{bmatrix} 0 & & & & & & & & & & \\ 1.1 & 0 & & & & & & & & & \\ 1 & 2.1 & 0 & & & & & & & & \\ 1 & 2 & 3.1 & 0 & & & & & & & \\ 1 & 2 & 3 & 4.1 & 0 & & & & & & \\ 1 & 2 & 3 & 4 & 5.1 & 0 & & & & & \\ 1 & 2 & 3 & 4 & 5 & 6.1 & 0 & & & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7.1 & 0 & & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8.1 & 0 & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9.1 & 0 & \end{bmatrix} \quad \text{(transpose)} .$$

Initial  $x = (1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1)^T$  with  $\varphi(x) = 38.09.$

Iteration	$\bar{t}$	$\bar{s}$	$t$	$s$	#QPs	$\varphi(x + d)$
1	1	0	1	0	1	37.08646
2	1	0	1	0	1	35.08646
3	1	0	1	0	1	31.08646
4	1	0	1	0	1	23.30168
5	1	1	1	1	1	23.06948
6	0	1	1	1	7	22.57218
7	1	1	1	1	3	22.55570
8	0	1	1	2	2	22.43732
9	0	1	1	3	3	22.39628
10	0	1	1	3	2	22.37459
11	1	2	1	2	1	22.37020
12	1	2	1	2	1	22.36642
13	1	2	1	2	1	22.36613
14	1	2	1	2	1	22.36612

Final  $x = (-21.25583, -20.58868, -19.24580, -18.60455, -17.22383, -16.63475, -15.18517, -14.74159, -13.05307, -13.46085)^T$  with

$$\lambda(x) = (22.36612, -17.32323, -20.48036, -21.34962, -21.69938, -22.17358, -22.26831, -22.33351, -22.36612, -22.36612)^T$$

and

$$U = [0.5], \quad V = \begin{bmatrix} 0.3445 & -5.017 \times 10^{-3} \\ -5.017 \times 10^{-3} & 0.1555 \end{bmatrix} .$$

*Comments.* This problem is quite difficult to solve, since at the solution the interior eigenvalues are nearly equal to  $\lambda_n$ . Indeed, if a larger value of TOL had been used, the QP probably would have become infeasible making it necessary to reduce TOL. During the first few iterations, larger improvements were inhibited by the trust region radius, which was successively doubled. At iteration 5 the QP solution indicated that  $t, s$  should be set to 1, 9, but since this would have made (4.10) greater than  $m + 1, t$  and  $s$  were not increased. As a result, seven QPs were required during iteration 6 until the trust radius was small enough to make progress. Eventually quadratic convergence was obtained once the correct multiplicities were identified. In this case the second derivative matrix  $W$  was essential for quadratic convergence.

In general we would not expect Step 3.2 to be required. The reason for this is that when  $t$  or  $s$  is increased to a value  $\geq 2$ , the iterate  $x$  is essentially moving onto a manifold which has dimension at least two lower than the current constraining manifold. This is unlikely to happen by accident, but only likely to occur in the course of making progress towards optimality. However, the ability to split multiple eigenvalues is still important in case it is needed because of starting at an unfortunate point or in the course of solving ill-conditioned problems.

**7. Final comments.** A number of problems in addition to (1.1) may be solved by related techniques. Clearly it is trivial to extend the algorithm given here to solve

$$\min_x \max_{1 \leq l \leq p} \max_{1 \leq i \leq n} |\lambda_i(A^{(l)}(x))|,$$

where  $A^{(1)}(x), \dots, A^{(p)}(x)$  are each affine matrix-valued functions, by simply introducing additional constraints to the QP and corresponding Lagrange matrices. The algorithm could also be extended to solve more general optimization problems involving constraints on eigenvalues of various matrix functions. It would be necessary to introduce a penalty function to measure progress towards the solution. Constraints on interior eigenvalues could also be included (although these would not be convex).

Finally, it is possible to extend the algorithm to handle nonlinear matrix functions  $A(x)$ , although the resulting optimization problem is no longer necessarily convex. The necessary changes are mainly to replace  $A_k$  by  $\partial A(x)/\partial x_k$  in the derivative formulas, and to be aware of the need to verify second-order optimality conditions.

#### REFERENCES

- J. CULLUM, W. E. DONATH, AND P. WOLFE (1975), *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Study, 3, pp. 35–55.
- J. DOYLE (1982), *Analysis of feedback systems with structured uncertainties*, IEEE Proc., 129, pp. 242–250.  
 ——— (1986), private communication.
- R. FLETCHER (1985), *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23, pp. 493–513.
- S. FRIEDLAND, J. NOCEDAL, AND M. L. OVERTON (1987), *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal., 24, pp. 634–667.
- P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT (1984), *User's Guide to QPSOL: a Fortran package for quadratic programming*, Systems Optimization Laboratory Report, Stanford University, Stanford, CA.
- W. MURRAY AND M. L. OVERTON (1980), *A projected Lagrangian algorithm for nonlinear minimax optimization*, SIAM J. Sci. Statist. Comput., 1, pp. 345–370.
- J. NOCEDAL AND M. L. OVERTON (1985), *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22, pp. 821–850.
- N. OLHOFF AND J. E. TAYLOR (1983), *On structural optimization*, J. Appl. Mech., 50, pp. 1138–1151.
- M. L. OVERTON (1983), *A quadratically convergent method for minimizing a sum of Euclidean norms*, Math. Programming, 27, pp. 34–63.
- E. POLAK AND Y. WARDI (1982), *Nondifferentiable optimization algorithm for designing control systems having singular value inequalities*, Automatica, 18, pp. 267–283.
- R. T. ROCKAFELLAR (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.  
 ——— (1981), *The Theory of Subgradients and Its Applications to Problems of Optimization: Convex and Nonconvex Functions*, Research and Education in Mathematics 1, Heldermann-Verlag, Berlin.
- B. T. SMITH, J. M. BOYLE, J. J. DONGARRA, B. S. GARBOW, Y. IKEBE, V. C. KLEMA, AND C. B. MOLER, (1967), *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Computer Science 6, Springer-Verlag, Berlin, New York.