Large-Scale Optimization of Eigenvalues

Michael L. Overton

Courant Institute of Mathematical Sciences
New York University

June 1991
Final Version for *SIAM J. Optimization*

Abstract

Optimization problems involving eigenvalues arise in many applications. Let x be a vector of real parameters and let $A(x)$ be a continuously differentiable symmetric matrix function of x. We consider a particular problem which occurs frequently: the minimization of the maximum eigenvalue of $A(x)$, subject to linear constraints and bounds on x. The eigenvalues of $A(x)$ are not differentiable at points x where they coalesce, so the optimization problem is said to be nonsmooth. Furthermore, it is typically the case that the optimization objective tends to make eigenvalues coalesce at a solution point.

There are three main purposes of the paper. The first is to present a clear and self-contained derivation of the Clarke generalized gradient of the max eigenvalue function in terms of a "dual matrix". The second purpose is to describe a new algorithm, based on the ideas of a previous paper by the author (SIAM J. Matrix Anal. Appl. 9 (1988) 256-268), which is suitable for solving large-scale eigenvalue optimization problems. The algorithm uses a "successive partial linear programming" formulation which should be useful for other large-scale structured nonsmooth optimization problems as well as large-scale nonlinear programming with a relatively small number of nonlinear constraints. The third purpose is to report on our extensive numerical experience with the new algorithm, solving problems which arise in the following application areas: the optimal design of columns against buckling; the construction of optimal preconditioners for numerical linear equation solvers; the bounding of the Shannon capacity of a graph. We emphasize the role of the dual matrix, whose dimension is equal to the multiplicity of the minimal max eigenvalue. The dual matrix is computed by the optimization algorithm and used for verification of optimality and sensitivity analysis.

1 Introduction

Eigenvalues of symmetric matrices play important roles in many different areas of applied mathematics. For perhaps the large majority of true applications, it is not the case that a fixed matrix, say A, is known, and its eigenvalues are needed. It is more typical that A depends on many parameters, and that the eigenvalues are desired for many different choices of the parameters. In many cases the choice of parameters is dictated by some optimization objective. For example, in a control application, where the size of the largest eigenvalue represents system stability, it may be desirable to minimize the largest eigenvalue, while in a structure analysis application, where the smallest eigenvalue represents a buckling load, it may be desirable to maximize the smallest eigenvalue. Other applications might have an optimization objective which does not involve eigenvalues (e.g. cost of a material), but include constraints on eigenvalues (e.g. ensure all eigenvalues are in a safe frequency interval).

In our work on optimization problems involving eigenvalues, we have found it very useful to concentrate on a particular model problem, namely to minimize the maximum eigenvalue of a symmetric $n \times n$ matrix $A(x)$, where $A(x)$ depends smoothly on a vector of parameters $x \in \Re^m$. It is useful and not significantly more complicated to allow the imposition of linear constraints on x. A common variation is to minimize the maximum eigenvalue in absolute value. (We avoid the term spectral radius, since this suggests complex eigenvalues; nonsymmetric matrices are not discussed in this paper, but see [OW88].) The model problem is directly applicable to many applications, including the first two mentioned above, while for other problems, e.g. those where only the constraints involve the eigenvalues, it is fairly clear how the main ideas should be extended.

The feature of eigenvalue optimization problems which makes them both particularly interesting and particularly difficult to solve is that the eigenvalues of a differentiable matrix function are not themselves differentiable at points where they coalesce. Furthermore, it is often the case that the optimization objective tends to make the eigenvalues coalesce at a solution point. For example, consider the model problem with

$$A(x) = \begin{bmatrix} 1 + x_1 & x_2 \\ x_2 & 1 - x_1 \end{bmatrix}.$$

The eigenvalues are

$$\lambda = 1 \pm \sqrt{x_1^2 + x_2^2}$$

so the maximum eigenvalue is minimized by $x = 0$. Clearly the maximum eigenvalue is not a smooth function at $x = 0$. More importantly, though, the max eigenvalue function cannot be written as the pointwise maximum of two smooth functions at $x = 0$; in other words, the eigenvalues themselves cannot be labeled, say, κ_1 and κ_2, each a smooth function of $x \in \Re^2$. Thus standard minmax optimization techniques (e.g. [MO80]) cannot be applied. Suggestions

for transforming the problem into a standard nonlinear programming form by means of determinants have been made [GT88], but these methods perform poorly [Pan89]; for other comments on the use of determinants, see [FNO87].

In the example given above, the maximum eigenvalue is convex in x. This is true in general when A depends linearly on x, since the Rayleigh principle can be used to show that the maximum eigenvalue is a convex function of the matrix elements. Because of this fact, it has been recognized for some time that the techniques of convex analysis (e.g. [Roc70]) are applicable to eigenvalue optimization problems; optimality conditions and/or first-order algorithms for various problem classes have been given by [CDW75,PW83,Doy82,Sha85b,Gol87,All89]. See also [OT83,Bra86] for discussion of problems arising in structural engineering.

In [Ove88], a quadratically convergent algorithm was given to solve the model problem, using a "dual matrix" formulation of the optimality conditions to fully exploit the nonsmooth problem structure. Two papers which greatly influenced this work were [FNO87,Fle85]. Numerical examples were given, demonstrating quadratic convergence to nonsmooth solutions. The assumption was made that $A(x)$ was affine, although it was indicated that this was not essential for the main ideas to apply. The reason for this is that the eigenvalues are nonsmooth, nonlinear functions of the matrix, so whether $A(x)$ depends linearly or nonlinearly on x is not of great importance, provided $A(x)$ is a smooth function. If $A(x)$ is nonlinear, the maximum eigenvalue is not necessarily convex in x, but it is a composition of a convex function with a smooth function. Optimality conditions for nonlinear $A(x)$, for the more general case of minimizing sums of largest eigenvalues (algebraically or in absolute value), are given by [OWa]. These optimality conditions are derived by characterizing Clarke's generalized gradient ([Cla83]) in terms of a dual matrix. Proofs of the local quadratic convergence of the successive quadratic programming algorithm used in [Ove88] are being developed in [OWb].

There are three main purposes of the present paper. The first is to present a clear and self-contained derivation of the generalized gradient of the max eigenvalue functional in terms of a dual matrix. An understanding of this is essential for the appreciation of the main ideas underlying our optimization algorithms. Our second contribution is to describe a new algorithm, based on the ideas of [Ove88], which is suitable for solving large-scale eigenvalue optimization problems. The third purpose of the paper is to report on our extensive numerical experience with the new algorithm, solving eigenvalue optimization problems which arise in three very interesting and quite different application areas.

The paper is organized as follows. Section 2 derives the generalized gradient of the max eigenvalue, and consequent optimality conditions for the model problem, using the dual matrix formulation. Section 3 discusses the role of the dual matrix in eigenvalue splitting and sensitivity analysis. Section 4 summarizes the eigenvalue optimization algorithm of [Ove88] and relates this to the generalized gradient derived in Section 2. Section 5 explains how to extend the main ideas of [Ove88] to solve problems with large numbers of variables. The

ideas of this section should also be useful for solving other structured large-scale nonsmooth optimization problems as well as nonlinear programming problems with a relatively small number of nonlinear constraints, both active areas of current research. Section 6 discusses how to efficiently compute the eigenvalues of the matrix iterates generated by the optimization algorithm when the dimension of the matrices is large. Section 7 explains how all of the foregoing may be generalized to apply to eigenvalue problems of the form $A(x)q = \lambda Bq$, where B is a fixed symmetric positive definite matrix. Section 8 discusses the case where several different matrix families are involved. Section 9 summarizes numerical results that have been obtained for a fascinating classical problem of Lagrange, finding the shape of the strongest column. Here the task is to maximize the smallest eigenvalue of a fourth-order differential equation. Section 10 discusses results obtained for finding optimal preconditioners for the solution of linear systems of equations. Section 11 discusses the application of our large-scale algorithm to a problem arising in graph theory, computing the Lovasz number of a graph. Section 12 makes some concluding remarks.

2 Optimality Conditions, the Generalized Gradient, and Dual Matrices

We start with some notation. Let $\Re^{n\times m}$ denote the set of n by m real matrices, and let $S\Re^{n\times n}$ denote the set of n by n real symmetric matrices. By $A \geq 0$, where A is symmetric, we mean that A is positive semi-definite. The notation $\| \, . \, \|$ will always denote the Euclidean vector norm. Let $\langle , \rangle$ denote the Frobenius inner product on the set of rectangular matrices, namely

$$\langle B, C \rangle \;=\; \mathrm{tr}\; B^T C = \mathrm{tr}\; C^T B = \sum b_{ij} c_{ij}$$

where the dimensions of the matrices depend on the context. (For example, B and C could be vectors.)

We now give a simple but important lemma.

Lemma 1 *The convex hull of the set*

$$\{ww^T : w \in \Re^n, \| \, w \, \| = 1\}$$

is the set

$$\{\tilde{U} : \tilde{U} \in S\Re^{n\times n}, \mathrm{tr}\; \tilde{U} = 1, \tilde{U} \geq 0\}.$$

Furthermore, the elements in the first set are the extreme points of the second set.

Proof. Any convex combination of first set is clearly contained in the second. Furthermore, any matrix in the second set has a spectral decomposition

$$\tilde{U} = \sum \theta_i w_i w_i^T$$

where the eigenvalues θ_i are nonnegative by the positive semi-definite condition and sum to one by the trace condition, and the eigenvectors w_i have unit norm, i.e. the right-hand side is a convex combination of elements in the first set. Clearly, any element of the first set is an extreme point of the second set. Also, any element of the second set which is not rank-one can be written as a nontrivial convex sum of elements in the first set and is therefore not an extreme point.

Theorem 1 *Let $A \in S\Re^{n\times n}$, and let $\lambda_1(A)$ be the largest eigenvalue of A. The following characterizations hold:*

$$\lambda_1(A) = \max\{\langle q, Aq\rangle \ : \| q \| = 1\}; \tag{1}$$

$$\lambda_1(A) = \max\{\langle qq^T, A\rangle \ : \| q \| = 1\}; \tag{2}$$

$$\lambda_1(A) = \max\{\langle \tilde{U}, A\rangle \ : \ \tilde{U} \in S\Re^{n\times n}, \ \mathrm{tr}\ \tilde{U} = 1, \ \tilde{U} \geq 0\}. \tag{3}$$

Consequently, λ_1 is a convex function of A.

Proof. Equation (1) is the well known Rayleigh quotient characterization. Equation (2) follows immediately from properties of the inner product. Equation (3) follows from Lemma 1, since maximizing a linear function over a set gives the same result as maximizing over its convex hull. The convexity follows from any of the characterizations, since the pointwise maximum of a set of linear functions is always convex. □

The characterization of a convex function as a pointwise maximum of a set of linear functions leads directly to the definition of the *subdifferential* of f. For example, suppose that $z \in \Re^k$, and

$$f(z) = \max\{\langle a_i, z\rangle + \beta_i \ : \ i \in \mathcal{I}\}$$

where $\mathcal{I}$ is a discrete index set. Then the subdifferential of f at z may be defined as

$$\partial f(z) = \mathrm{conv}\{a_i : i \in \mathcal{I}, \ f(z) = \langle a_i, z\rangle + \beta_i\}$$

where "conv" denotes convex hull. An important property of ∂f which immediately follows from this definition is that z minimizes f if and only if $0 \in \partial f(z)$; note also that f is differentiable at z if and only if the subdifferential contains only one element, namely the gradient of f at z. It is a fact [Roc70, Corollary 23.5.3] that the subdifferential may be defined in this way for general convex functions, giving, as a consequence of (2),

$$\partial\lambda_1(A) = \mathrm{conv}(\{qq^T \ : \ q \text{ is a normalized eigenvector for } \lambda_1(A)\}). \tag{4}$$

This leads to:

Theorem 2 *Suppose the maximum eigenvalue $\lambda_1(A)$ has multiplicity t, i.e. the eigenvalues of A are*

$$\lambda_1 = \cdots = \lambda_t > \lambda_{t+1} \geq \cdots \geq \lambda_n.$$

Then the subdifferential of λ_1 at A is the set

$$\partial\lambda_1(A) = \mathrm{conv}(\{Q_1 w w^T Q_1^T : w \in \Re^t, \| w \| = 1\}), \tag{5}$$

where the columns of Q_1 form an orthonormal set of eigenvectors for $\lambda_1(A)$. Another equivalent form is

$$\partial\lambda_1(A) = \{\tilde{U} = Q_1 U Q_1^T : \ U \in S\Re^{t\times t}, \ \mathrm{tr}\ U = 1, \ U \geq 0\}. \tag{6}$$

Proof. Equation (5) follows directly from (4), and (6) then follows from Lemma 1. Alternatively, writing the eigendecomposition of A as

$$A = Q\ \mathrm{Diag}(\lambda_i)\ Q^T, \quad Q = [Q_1\ Q_2],$$

we see that (6) follows from directly applying the definition of the subdifferential to (3) since the matrices on the right-hand side of (6) are those which achieve the maximum in (3), with

$$\tilde{U} = Q \begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix} Q^T.$$

No convex hull operation is necessary since the set is already convex. □

We now change notation, introducing $A(x) \in S\Re^{n\times n}$, a continuously differentiable function of $x \in \Re^m$, with eigenvalues

$$\lambda_1(x) \geq \cdots \geq \lambda_n(x),$$

and partial derivatives

$$A_k(x) = \frac{\partial A}{\partial x_k}(x).$$

It is convenient to use the symbol λ_1 for two purposes, with

$$\lambda_1(x) \equiv \lambda_1(A(x)),$$

and the distinction should be clear from the context. The function $\lambda_1(x)$ is not generally convex, but it is the composition of the convex function $\lambda_1(A)$ with the smooth function $A(x)$. The Clarke *generalized gradient* of $\lambda_1(x)$ may therefore be defined by means of a chain rule [Cla83, p.42], [Fle87, p.366]. We obtain:

Theorem 3 *Suppose the maximum eigenvalue of $A(x)$ has multiplicity t, with a corresponding orthonormal basis of eigenvectors $Q_1(x) = [q_1(x), \ldots, q_t(x)]$. The generalized gradient of $\lambda_1(x)$ is the set*

$$\partial\lambda_1(x) = \{v \in \Re^m \ : \ v_k = \ \langle U, Q_1(x)^T A_k(x) Q_1(x)\rangle, \tag{7}$$

$$\text{for some } U \in S\Re^{t\times t}, U \geq 0, \mathrm{tr}\ U = 1\}.$$

Proof. By the chain rule just cited,

$$\partial\lambda_1(x) = \{v \in \Re^m \ : \ v_k = \ \langle G, A_k(x)\rangle \ \text{ for some } G \in \partial\lambda_1(A)\}.$$

The proof is completed by using (6) and noting that

$$\langle Q_1 U Q_1^T, A_k\rangle \ = \ \langle U, Q_1^T A_k Q_1\rangle. \square$$

Equation (4) is well known; see [CDW75,PW83,Cla83]. The equivalent form (6) is much less known and much more useful, as we shall see shortly; the earliest reference we know for this explicit form is Fletcher [Fle85], where a different proof was given. Equation (7) was given in the case that $A(x)$ is affine in [Ove88], using a proof based on Fletcher's work. The proofs given here make more use of the machinery of [Cla83,Roc70]. A referee has pointed out that Clarke's powerful theory is not required for Theorem 3 and subsequent results, which could in fact be obtained from the theory of "locally convex" functions; see [IT79,Sha85b]. We prefer to refer to Clarke's work so that we may use the beautifully simple notion of a chain rule developed there.

The matrix $\tilde{U}$ may be viewed as a "dual matrix"; indeed, a "dual problem" is formulated at the end of this section. The $t \times t$ matrix U may be called a "reduced dual matrix", but since it is the one we shall need as a computational tool we shall also refer to it as simply the dual matrix. (The term "Lagrange matrix" was used in [Ove88].) The distinction between $\tilde{U}$ and U is analogous to the notational question of whether inactive constraints in a nonlinear program should be assigned zero Lagrange multipliers.

Theorem 3 gives a form of the generalized gradient which is particularly useful for computation, since it does not involve taking a convex hull. Indeed, it characterizes the generalized gradient using *structure functionals*, to use a term introduced by Osborne [Osb85] for some other nonsmooth optimization problems. In our case, the structure functionals may be taken to be the $t(t+1)/2$ quantities

$$q_i^T A(x) q_j, \quad 1 \le i \le j \le t, \tag{8}$$

assuming the eigenvectors $q_1, \ldots, q_t$ are fixed. Theorem 3 then states that the generalized gradient of $\lambda_1(x)$ consists of particular linear combinations of the gradients of the structure functionals, namely those with coefficients u_{ii} and $2u_{ij}$ $(j \neq i)$ making up a positive semi-definite dual matrix U with trace one. (A better definition of the structure functionals, which would allow statements about second-order effects, would presumably use the matrix exponential formulation mentioned in Section 4.)

Note that the eigenvector basis Q_1 for $\lambda_1(x)$ is not unique if $t > 1$ (and even if $t = 1$ the sign is not unique). However, replacing Q_1 by any other valid choice, which must have the form $Q_1 V$ for some $t \times t$ orthogonal matrix V, simply transforms the dual matrix U into VUV^T, preserving its eigenvalues.

The directional derivative of λ_1 is easily deduced from the generalized gradient formula. We have

Theorem 4 *Under the assumptions of Theorem 3, the directional derivative*

$$\lambda_1'(x;d) = \lim_{\alpha \to 0+} \frac{\lambda_1(x+\alpha d) - \lambda_1(x)}{\alpha}$$

is the largest eigenvalue of

$$B(d) = \sum_{k=1}^{m} d_k Q_1(x)^T A_k(x) Q_1(x). \tag{9}$$

Proof. Because $\lambda_1(x)$ is the composition of a convex function with a smooth function,

$$\lambda_1'(x;d) = \max_{v \in \partial \lambda_1(x)} \langle v, d \rangle$$

(see [Fle87, p.369] or [Cla83, Ch.2]). By (7), we therefore obtain

$$\lambda_1'(x;d) = \max_{U} \langle U, B(d) \rangle$$

where the max is taken over positive semi-definite matrices with trace one. The result therefore follows from Theorem 1. □

The formula for the directional derivative may alternatively be obtained from the classical results in [Kat82], which state that the multiple eigenvalue $\lambda_1 = \cdots = \lambda_t$ of $A(x)$ splits into t eigenvalues of $A(x+\alpha d)$, for α near 0, with corresponding derivatives equal to the eigenvalues of $B(d)$. However, the proof of this basic fact is not at all straightforward, especially in the case that $A(x)$ cannot be extended to an analytic function of complex variables.

We now consider optimality conditions for a constrained version of the model problem.

Theorem 5 *Consider the problem:*

$$\min_x \lambda_1(x) \tag{10}$$

subject to

$$Cx = b; \quad \ell \le x \le u \tag{11}$$

where $C = [c_1, \ldots, c_m] \in \Re^{n_c \times m}, b \in \Re^{n_c}, \ell$ *and* $u \in \Re^m$. *Then a necessary condition for* x *to solve (10)–(11) is, in addition to (11), that there exists a dual matrix* $U \in S\Re^{t \times t}$, *where* t *is the multiplicity of* $\lambda_1(x)$, *and vectors of Lagrange multipliers* $\mu \in \Re^{n_c}$ *and* $\gamma \in \Re^m$, *satisfying*

$$\langle U, Q_1(x)^T A_k(x) Q_1(x) \rangle \;=\; \langle \mu, c_k \rangle \;+\; \gamma_k, \quad k = 1, \ldots, m \tag{12}$$

$$\text{tr } U = 1 \tag{13}$$

$$U \ge 0 \tag{14}$$

and

$$\gamma_k = 0 \text{ if } \ell_k < x_k < u_k;\ \gamma_k \geq 0 \text{ if } x_k = \ell_k;\ \gamma_k \leq 0 \text{ if } x_k = u_k. \qquad (15)$$

Here the columns of $Q_1(x)$ form an orthonormal basis of t eigenvectors for $\lambda_1(x)$. The necessary condition (together with the satisfaction of (11)) is also sufficient for optimality if $A(x)$ is affine.

Proof. It follows from the standard Lagrange multiplier rule for nonsmooth optimization [Cla83, p.228], which reduces to $0 \in \partial\lambda_1(x)$ in the case that there are no constraints. The last statement holds because if $A(x)$ is affine, $\lambda_1(x)$ is a composition of a convex with an affine function, and therefore convex. □

We complete this section with a discussion of a duality result, which clarifies the terminology "dual matrix". By (3), the "primal problem" (10)–(11) is equivalent to

$$\min_{Cx=b;\ \ell\leq x\leq u} \quad \max_{\tilde{U}:\ \mathrm{tr}\,\tilde{U}=1,\ \tilde{U}\geq 0} \langle \tilde{U}, A(x)\rangle.$$

(Here, as before, $x \in \Re^m$ and $\tilde{U} \in S\Re^{n\times n}$.) Now define a "dual problem"

$$\max_{\tilde{U}:\ \mathrm{tr}\,\tilde{U}=1,\ \tilde{U}\geq 0} \quad \min_{Cx=b;\ \ell\leq x\leq u} \langle \tilde{U}, A(x)\rangle.$$

The following theorem, motivated originally by [Boy87,OWa], is a standard saddle point result and follows from [Roc70, Theorem 36.3]. For closely related results, see [Ehr79,Sha85a].

Theorem 6 *Suppose that $A(x)$ is an affine function, so that $A_k(x)$ is constant (independent of x) for all k. If the primal problem has a solution, say defined by $(x^*, \tilde{U}^*)$, then the same pair solves the dual problem.*

Note that in the unconstrained affine case the dual problem can have a solution $\tilde{U}$ with corresponding objective greater than $-\infty$ only if

$$\langle \tilde{U}, A_k\rangle \ = \ 0, \quad k = 1, \ldots, m.$$

Consequently, the dual version of the unconstrained affine primal problem is

$$\max\{\langle \tilde{U}, A(0)\rangle \ :\ \mathrm{tr}\ \tilde{U} = 1,\ \tilde{U} \geq 0,\ \langle \tilde{U}, A_k\rangle = 0,\ k = 1, \ldots, m\}. \qquad (16)$$

3 Eigenvalue Splitting and Sensitivity Analysis

The following theorem shows the importance of the eigenvalues of the $t \times t$ dual matrix U.

Theorem 7 *Suppose that x, U, μ and γ satisfy all the conditions (11)–(15) except possibly the semi-definite condition (14), and let κ be an eigenvalue of U with corresponding normalized eigenvector $v \in \Re^t$. If $d \in \Re^m, \delta \in \Re$ satisfy the following linear system of equations,*

$$\sum_{k=1}^{m} d_k Q_1^T A_k(x) Q_1 - \delta I = -vv^T \tag{17}$$

$$Cd = 0 \tag{18}$$

$$d_k = 0 \text{ if } x_k = \ell_k \text{ or } x_k = u_k. \tag{19}$$

then d is a feasible direction with directional derivative

$$\lambda_1'(x; d) = \kappa.$$

Proof. It is clear that d is a feasible direction. The eigenvalues of the first matrix term on the left-hand side of (17) are, by construction, all equal to δ except one which has the value $\delta - 1$. It follows from Theorem 4 that the desired directional derivative has the value δ. Taking an inner product of U with both sides of (17) yields, using (12),

$$\sum_{k=1}^{m} d_k(\langle \mu, c_k \rangle + \gamma_k) - \delta = -\kappa,$$

i.e.

$$\mu^T C d + \gamma^T d - \delta = -\kappa$$

which gives, using (18)–(19) and (15),

$$\delta = \kappa. \quad \square$$

This theorem was given in the unconstrained affine case by [Ove88]. It was also explained there that for unconstrained problems, the multiplicity t of the multiple eigenvalue λ_1 is generically restricted by

$$\frac{t(t+1)}{2} \leq m + 1, \tag{20}$$

the right-hand side being regarded as the number of degrees of freedom available. (The "1" reflects the fact that the value of the multiple eigenvalue is free.) This restriction is known as the von Neumann-Wigner crossing rule and is well known

in quantum mechanics; it is further motivated in [FNO87]. For problems with the linear constraints and bounds (11) it is clearly necessary to replace (20) by

$$\frac{t(t+1)}{2} \leq m + 1 - n_c - n_b, \tag{21}$$

where n_b is the number of active bounds, i.e. the number of variables x_k which are equal to either ℓ_k or u_k. Note, then, that with this nondegeneracy assumption on t, the linear system (17)–(19), which consists of $t(t+1)/2 + n_c + n_b$ linear equations in $m+1$ variables, is generically solvable.

Theorem 7 shows how a descent direction may generically be computed in the event that a point x satisfies all the optimality conditions except the positive semi-definite condition on U. This direction splits the multiple eigenvalue into two clusters, one of unit multiplicity and one of multiplicity $t-1$, to first order. (See the discussion following Theorem 4.) Clearly, other splitting choices are possible; the one given here may be regarded as a generalization of the standard procedure for moving off constraints associated with multipliers of the wrong sign in linear or nonlinear programming, namely moving off only one constraint at one time. Note that the coefficient matrix of the linear system (17)–(19) is the transpose of the coefficient matrix describing the active optimality conditions (12), (13), (15).

Theorem 7 also shows how the eigenvalues of the dual matrix U describe the sensitivity of an optimal solution along directions which split the multiple eigenvalue λ_1 to first order. In particular, the theorem shows how to quantify first-order changes in λ_1 along these directions. If equality holds in (21), then, generically, all feasible directions in $\Re^m$ split the multiple eigenvalue to first order; in this case an optimal solution is characterized by first-order information and is generically "strongly unique". However, (21) cannot usually be expected to hold with equality, in which case there exists a nontrivial subspace of feasible directions d along which λ_1 does not split to first order, i.e. feasible directions tangent to the nontrivial manifold along which the eigenvalue retains multiplicity t. Since the function λ_1 is smooth along this manifold, it exhibits only second-order changes away from an optimal point x along these directions. The magnitude of these second-order changes is determined by the eigenvalues of the appropriate reduced Lagrangian Hessian, just as in nonlinear programming.

4 The Successive Quadratic Programming Algorithm

Let x^* be a local minimum of $\lambda_1(x)$; if $A(x)$ is affine, x^* is also a global minimum. Suppose that $\lambda_1(x^*)$ has multiplicity t^*. We wish to generate a sequence of iterates x^ν converging to x^*, but even if $t^* > 1$, usually $A(x^\nu)$ has distinct eigenvalues for any finite value of ν. (A similar remark applies to nonlinear programming problems; nonlinear constraints are generally both active and satisfied only in the limit.) In order for an algorithm to have good convergence

properties, therefore, it is important for it to exploit the structure of the generalized gradient which is estimated to apply at the limit point, not just the gradient information at the current iterate. This observation is the basis for the so-called "ϵ -subgradient" methods found in [LM78], and similarly it is the estimated optimal active constraint structure which underlies successive quadratic programming (SQP) methods for nonlinear programming. In the latter case this estimated structure is usually defined by the active set found at the solution of the approximating quadratic program.

The algorithm presented in [Ove88] takes full advantage of the structure of the generalized gradient which is estimated to apply at the optimal point. To do so, it requires an estimate of t^*, say t, which is obtained and revised as the algorithm proceeds. One way of doing this was suggested in [Ove88], but more recent numerical experience suggests that a simpler approach is better. Let x be the current iterate, with $A(x)$ having eigenvalues

$$\lambda_1(x) \geq \cdots \geq \lambda_n(x),$$

with a corresponding orthonormal set of eigenvectors $\{q_i(x)\}$, and define t in terms of a tolerance τ by

$$\lambda_1(x) - \lambda_t(x) \leq \tau \max(1, |\lambda_1(x)|); \quad \lambda_1(x) - \lambda_{t+1}(x) > \tau \max(1, |\lambda_1(x)|). \tag{22}$$

Define

$$Q_1(x) = [q_1(x), \ldots, q_t(x)]. \tag{23}$$

It will usually be necessary to adjust τ during the course of the minimization process.

The basic iteration of the method of [Ove88] is defined by solving the following quadratic program (QP):

$$\min_{d,\delta} \delta + d^T W d \tag{24}$$

subject to

$$\delta I - \sum d_k Q_1(x)^T A_k(x) Q_1(x) = \mathrm{Diag}(0, \lambda_2(x) - \lambda_1(x), \ldots, \lambda_t(x) - \lambda_1(x)) \tag{25}$$

$$\delta - \sum d_k q_i(x)^T A_k(x) q_i(x) \geq \lambda_i(x) - \lambda_1(x), \quad i = t+1, \ldots, n \tag{26}$$

$$\| d \|_\infty \leq \rho, \tag{27}$$

where d and δ are variables in $\Re^m$ and $\Re$ respectively, W is a positive definite matrix, and ρ is a trust region radius updated by the algorithm.

The motivation for the constraint (25) is that it results from linearizing a differentiable system of $t(t+1)/2$ nonlinear equations characterizing the condition $\lambda_1(x) = \cdots = \lambda_t(x) = \omega$, for some $\omega \in \Re$. Actually, as was pointed out by [Zho88], the form of the nonlinear system given by (4.1) of [Ove88] is not

correct. The correct system uses a matrix exponential formulation based on Theorem 3.1 of [FNO87], as is explained in more detail in [OWb]. Constraints (26) ensure that linearizations of $\lambda_{t+1}, \ldots, \lambda_n$ give values no greater than the linearized value for the approximate multiple eigenvalue $\lambda_1, \ldots, \lambda_t$. Both (26) and (27) prevent d from having too large norm, particularly during the early part of the iteration. Ideally, they will not be active near the solution.

The constraint (25) is imposed as $t(t+1)/2$ scalar constraints, each of which has a QP multiplier associated with it. These multipliers make up the QP dual matrix estimate U, with diagonal elements of U equal to the corresponding multipliers for the diagonal equations in (25) and off-diagonal elements of U equal to half the corresponding multipliers for the off-diagonal equations in (25).

Constraints on the variables were not considered in [Ove88] for simplicity, but let us explicitly include linear constraints and bounds in the present discussion, i.e. address the problem (10)–(11). Assume that the present iterate x satisfies (11); then the corresponding restrictions which should be added to the QP are

$$Cd = 0 \tag{28}$$

$$\ell \leq x + d \leq u. \tag{29}$$

The following theorems clarify some points that were not made in [Ove88].

Theorem 8 *Suppose the quadratic program (24)–(29) has solution d, δ with the property that constraints (26)–(27) are not active. Then the solution has an associated dual matrix U and vectors of multipliers μ and γ satisfying*

$$(Wd)_k \;+\; \langle U, Q_1^T A_k(x) Q_1 \rangle \;=\; \langle \mu, c_k \rangle \;+\; \gamma_k, \quad k = 1, \ldots, m \tag{30}$$

$$\text{tr } U = 1 \tag{31}$$

and

$$\gamma_k = 0 \text{ if } \ell_k < x_k + d_k < u_k;\ \gamma_k \geq 0 \text{ if } x_k + d_k = \ell_k;\ \gamma_k \leq 0 \text{ if } x_k + d_k = u_k. \tag{32}$$

Furthermore, U, μ and γ are unique if the $t(t+1)/2 + n_c$ linear constraints (25), (28) on d, δ, together with the active bound restrictions on d, are linearly independent.

Proof. It follows immediately from the standard optimality conditions for quadratic programs (see e.g. [GMW81]). □

Theorem 9 *Assume $\tau = 0$, so that $\lambda_1(x)$ has exact multiplicity t. The quadratic program (24)–(29) yields a vector d which is a descent direction for λ_1, unless $d = 0$. Furthermore, if $\rho > 0$, then $(d = 0, \delta = 0)$ solves the QP if and only if (12), (13) and (15) are satisfied for some U, μ and γ, i.e. the optimality conditions (12)–(15) are satisfied, with the possible exception of the positive semi-definite condition on U.*

Proof. By (25) combined with Theorem 4, we have

$$\lambda_1'(x;d) = \delta \tag{33}$$

Also, the QP solution (d, δ) satisifies

$$\delta + \frac{1}{2} d^T W d \leq 0$$

since the value zero is achievable with $(d = 0, \delta = 0)$. Thus

$$\lambda_1'(x;d) \leq -\frac{1}{2} d^T W d.$$

Since W is positive definite, the right-hand side is nonpositive, with zero value if and only if $d = 0$. The last statement follows from Theorem 8. □

If it happens that $d = 0$, so that the optimality conditions are satisfied with the possible exception of the positive semi-definite condition on U, and if indeed U has a negative eigenvalue, then it is necessary to split the multiple eigenvalue $\lambda_1(x)$ as explained in Theorem 7 in order to obtain a decrease in the maximum eigenvalue. In nonlinear programming, an analogous situation occurs when x satisfies all optimality conditions except the sign constraints on the Lagrange multipliers.

Whether d is zero or not, equations (30)–(31) define a matrix U which is unique as long as the active constraint gradients of the QP are linearly independent. (Note that (21) is a necessary condition for such independence.) If the dual matrix estimate U generated by the QP is not positive semi-definite, this is a clear indication that the multiplicity estimate t is too large and that the tolerance τ should be reduced if possible. This strategy is used in the current version of our programs. Consequently, we do not generally expect to converge to points x where it is necessary to split a multiple eigenvalue. This is indeed the case in practice, with the notable exception of the graph problems to be described in Section 11.

Theorem 10 *Suppose that the QP (24)–(29) yields a solution d, δ with the property that the constraints (26) are not active, and suppose that U defined by (30)–(31) is positive semi-definite. Then d is a descent direction for λ_1 (unless $d = 0$), regardless of the value of τ.*

Proof. The exact multiplicity of $\lambda_1(x)$ is less than or equal to the multiplicity t defined by (22). Consequently, (33) holds, just as in Theorem 9. However, $(d = 0, \delta = 0)$ does not generally satisfy (25). Let

$$E\bar{d} = e$$

represent the combined linear system (25) and (28), where $\bar{d} = (d^T, \delta)^T$. It follows that equations (30)–(31) may be written

$$\begin{bmatrix} Wd \\ 1 \end{bmatrix} = E^T v + \begin{bmatrix} \gamma \\ 0 \end{bmatrix}$$

where $v = (U_{11}, 2U_{12}, \cdots, U_{tt}, \mu_1, \cdots, \mu_{n_e})$. (Actually, this system needs modification if the trust radius constraint (27) is active, but this is easily done by modifying the corresponding lower and upper bounds ℓ_k or u_k to impose the trust radius bound.) Taking an inner product with $\bar{d}$ we have

$$d^T W d + \delta = v^T e + \gamma^T d.$$

We have $\gamma^T d \leq 0$ by (32) together with feasibility of x. Since e has non-positive entries corresponding to diagonal elements of U in v and zero entries elsewhere, we therefore have

$$\delta \leq 0$$

from the semi-definiteness of U and W, with $\delta = 0$ only if $d = 0$ (since W is positive definite). □

It follows that if the dual matrix estimate U is positive semi-definite and τ and ρ are both sufficiently small,

$$\lambda_1(x + d) < \lambda_1(x) \tag{34}$$

provided d is nonzero. (If τ is too large relative to ρ the QP may not be feasible, while if ρ is too large, $\| d \|$ may be too large for the negative directional derivative to yield (34).) The best automatic way to adjust τ and ρ is not clear but in practice, given a reasonable estimate for τ, obtaining the reduction (34) by decreasing ρ is usually straightforward unless λ_1 is very near its optimal value. Provided (34) holds, the new iterate may be set to $x + d$. (The difficulty of a possibly infeasible subproblem is eliminated in the large-scale algorithm described in the next section.)

It is explained in [Ove88] that, in order to obtain a quadratically convergent method, W should be set to the Hessian of the appropriate Lagrangian function. We emphasize that W is not the Hessian of the max eigenvalue function, which does not exist at x^* if $t^* > 1$. The correct form of the Lagrangian is not (4.9) of [Ove88] but a modification using the matrix exponential formulation mentioned above. The formula for W given by (4.12) of [Ove88] is correct. Its derivation was omitted, but it is given in [OWb]. In the case $t = 1$, the formula reduces to a fairly well known expression for the second derivative of a distinct eigenvalue; see [Lan64,GV83]. In the case that $A(x)$ is nonlinear, an additional term

$$Q_\ell^T \frac{\partial^2 A}{\partial x_j \partial x_k} Q_\ell$$

must be added to (4.12) of [Ove88], assuming that $A(x)$ is twice continuously differentiable.

We make here an observation not made in [Ove88], namely that in some cases the reduction condition (34) may not hold for ρ large enough that (27) is inactive, even when W is set to the correct Hessian matrix and x is very close to an optimal solution. Such a situation is known as the Maratos effect

and it prevents quadratic convergence of the algorithm, since the trust radius ρ must be reduced until it yields (34). This difficulty has indeed occurred on some of our test problems, but it has been overcome by implementing Fletcher's second-order correction technique, making use of our knowledge of the Hessian matrix W to avoid additional gradient evaluations as does Fletcher in [Fle85].

Clearly, it is important to develop a precise version of the algorithm for which global convergence can be guaranteed. As yet we have not attempted to do this, but we do not see any inherent difficulty. Trust region convergence proofs are by now rather well understood; the essential ingredients in this case are given by the theorems above.

The SQP algorithm summarized in this section has been used to solve a wide variety of problems, some of which will be mentioned in later sections of the paper. Our Fortran implementations use Eispack subroutines [S$^+$67] to obtain the eigenvalues and eigenvectors of each matrix $A(x)$ and either the Stanford code LSSOL [GMSW86] or the equivalent NAG routine [NAG] to solve the quadratic programs. Using current workstation technology, only a moderate amount of computer time is typically required to obtain a very accurate solution, including verification of the optimality conditions, for, say, $\max(n, m) \leq 40$. However, the algorithm is very inefficient for much larger values of n, m. The next two sections discuss how to modify the algorithm for large-scale problems.

5 The Optimization Algorithm when m is Large

In this section we discuss our approach to modifying the successive quadratic programming algorithm when m is large, say $m > 40$. The discussion of how to efficiently compute the eigenvalues when n is also large is deferred to the next section.

The first observation is that the benefits of quadratic convergence are far outweighed by the cost of computing and factoring the Lagrangian Hessian W when m is large. We shall therefore consider a first-order algorithm based on successive linear programming instead of successive quadratic programming, replacing W by zero in (24). First-order algorithms, which generally converge at a first-order rate, can be very satisfactory in some applications; in other cases they can be excruciatingly slow. If it happens that equality holds in (21) then, generically, the solution is "strongly unique", which implies that a first-order method is quadratically convergent. However, this is not generally to be expected.

A successive linear programming method retains the key feature of the SQP algorithm of [Ove88], namely that the algorithm estimates the eigenvalue multiplicity t and uses the appropriate $t(t+1)/2$ linear constraints to approximate the condition $\lambda_1(x+d) = \cdots = \lambda_t(x+d) = \omega$, generating the corresponding $t \times t$ dual matrix U. Consequently, verification of the optimality conditions for the model problem is possible. We have:

Theorem 11 *Assume that $\tau = 0$, so that $\lambda_1(x)$ has exact multiplicity t. Then the linear program (24)–(29), where $W = 0$, yields a vector d which is a direction of non-ascent for $\lambda_1(x)$. Furthermore, if $\rho > 0$ then $(d = 0, \delta = 0)$ is a (not necessarily unique) solution of the linear program if and only if (12), (13) and (15) are satisfied for some U, μ, and γ.*

Proof. It is a straightforward modification of the proof of Theorem 9. The solution $(d = 0, \delta = 0)$ cannot be unique when $t(t+1)/2 + n_c + n_b < m + 1$ since it is not a vertex of the feasible region. □

However, even solving the linear program (24)–(29), where $W = 0$, is not a justifiable expense when m is large, especially if $t(t+1)/2 + n_c + n_b << m$, which is usually the case. Usually the LP has only a few active general linear constraints, namely (25) and (28), so that obtaining a vertex solution requires most of the elements of d to be on their bounds. Often, aside from perhaps a few "genuine" active bounds arising in (29), most of the active bounds are trust radius bounds in (27). If the simplex method is used to solve the LP, most of the work involves finding the active set of bounds. Since there are only a few general linear constraints, the work per simplex step need only be $O(m)$, but $O(m)$ steps are required. This is not acceptable, especially since the exact set of active trust radius bounds is of little importance; the purpose of the trust radius is simply to restrict d so that its norm is not too large.

In view of these remarks we have implemented the following "partial linear programming" solver. (For a related idea, see [GO89].)

PLP Algorithm to partially solve the LP

$$\min g^T \bar{d} \tag{35}$$

subject to

$$E\bar{d} = e \tag{36}$$

$$F\bar{d} \geq f \tag{37}$$

$$\bar{d}_k = 0, \quad k \in K \tag{38}$$

$$\bar{\ell} \leq \bar{d} \leq \bar{u} \tag{39}$$

$$\| \bar{d} \|_\infty \leq \rho, \tag{40}$$

where $\bar{d} = (d^T, \delta)^T \in \Re^{m+1}$, K is an index set, and $g, E, e, F, f, \bar{\ell}, \bar{u}$ are defined so that (35)–(40) is equivalent to (24)–(29), with $W = 0$, except that the additional constraints (38) have been introduced (for reasons to be explained shortly) and that, for convenience, the trust radius restriction applies to $\bar{d}$ instead of d. Thus

$$g = [0, \ldots, 0, 1]^T;$$

E and e respectively contain the $t(t+1)/2$ rows

$$[-q_i(x)^T A_1(x) q_j(x), \ldots, -q_i(x)^T A_m(x) q_j(x), \delta_{ij}]; \;\; \delta_{ij}(\lambda_i(x) - \lambda_1(x)),$$

$1 \leq i \leq j \leq t$ (where δ_{ij} is the (i,j) element of the identity matrix), together with the additional n_c rows

$$[C \;\; 0]; \quad 0;$$

F and f contain the rows

$$[-q_i(x)^T A_1(x) q_i(x), \ldots, -q_i(x)^T A_m(x) q_i(x), 1]; \quad \lambda_i(x) - \lambda_1(x),$$

$i = t+1, \ldots, n$; and

$$\bar{\ell} = [(\ell - x)^T, -\infty]^T; \;\; \bar{u} = [(u - x)^T, \infty]^T.$$

It is assumed that $\ell \leq x \leq u$, so that $\bar{\ell} \leq 0$, $\bar{u} \geq 0$. Note also that $f \leq 0$, so $\bar{d} = 0$ satisfies all constraints except (36). It is assumed that $t(t+1)/2 + n_c << m$. It is not necessary to store or even fully compute the derivative matrices $A_k(x)$; rather, a subroutine is required to perform the matrix vector product $A_k(x)q$ for given index k and vector q.

Step 0. Set $\nu = 0$. Set $\bar{d}^0$ to the least-norm solution of the underdetermined linear system (36), (38). This is obtained by a QR factorization of G, a matrix defined initially to contain the columns of E^T, with rows corresponding to the indices in K removed. Let the QR factorization of G be given by

$$G = YR$$

where R is upper triangular and Y, which has the same dimensions as G, satisfies $Y^T Y = I$. Then solve

$$R^T d_Y = e$$

and set

$$\tilde{d}^0 = Y d_Y, \tag{41}$$

the least norm solution of $G^T \tilde{d} = e$. Set $\bar{d}^0$ to the vector containing $\tilde{d}^0$ interspersed with zeros corresponding to the entries in K. (We use the Linpack software for computing the QR factorization; the range space basis Y is stored only as a product of Householder transformations. For details see [D$^+$78] and, for information on how to update the factorization and use it in the context of optimization, [GMW81].) Then set $\bar{d} = \alpha^0 \bar{d}^0$, where α^0 is defined as follows. If $\bar{d}^0$

is a feasible point for the LP, set $\alpha^0 = 1$. Otherwise, if $\bar{d}^0$ violates the constraints (37), the bounds (39) or the trust radius restriction (40), set α^0 to the maximum value possible so that $\bar{d}$ satisfies (37)–(40). (This effectively modifies the equality constraints of the LP. The rationale here is that if the least norm step to the equality constraints of the LP is infeasible, most likely the approximations underlying the definition of the LP are not good enough to justify its solution, should it indeed have a feasible solution.)

Step 1. Let $\tilde{g}$ be g with rows corresponding to the indices in K removed. Set $\tilde{d}$ to the least-squares projection of $\tilde{g}$ onto the null space of G^T. This is obtained by using the QR factorization of G to solve the least squares problem

$$\min_v \| Gv - \tilde{g} \|_2, \tag{42}$$

i.e. solving

$$Rv = Y^T \tilde{g},$$

and setting $\tilde{d}$ to the residual $Gv - \tilde{g}$. Note that a null space basis is *not* computed. If $\| \tilde{d} \| \leq \epsilon$, go to Step 3. Otherwise increment ν, and set $\bar{d}^\nu$ to the vector containing $\tilde{d}$ interspersed with zeros corresponding to the entries in K.

Step 2. Compute the maximum step α^ν so that

$$\bar{d} = \sum_{\kappa=0}^{\nu} \alpha^\kappa \bar{d}^\kappa \tag{43}$$

satisfies the constraints of the LP, consequently making a new general linear constraint or bound active. In the former case, append the corresponding row of F as a new column of G. In the latter case, if the new active bound is one of the bounds in (39), add the corresponding index to K and remove the corresponding row from G. In either of these cases, update the QR factorization of G accordingly and go back to Step 1. Finally, if the new active bound is one of the bounds in (40), go to Step 3.

Step 3. Set v to the final vector of constraint multipliers, by permuting the elements of the last solution of (42) to correspond to the row order in E and F, interspersing zeros corresponding to inactive constraints in (37). Set γ to the final vector of bound multipliers, by setting

$$\gamma_k = (g - [E^T \ F^T]v)_k, \quad k \in K$$

and $\gamma_k = 0$ otherwise. (See [GMW81], p. 189.) Exit with $\bar{d}$ (defined by (43)) , v, γ and K.

The basic idea of the PLP algorithm is that once one active trust radius bound is encountered, there is little to be gained by going through the computationally expensive process of adding all the other active trust radius bounds making up a vertex solution to the LP. Of course, since the PLP method neither checks multiplier signs nor allows a constraint or bound, once active, to become inactive, it will not generally produce an optimal solution of the LP.

Note that when $t = 1$, $\bar{d}^0 = 0$ and the vector consisting of the first m components of $\alpha^1 \bar{d}^1$, say $\alpha^1 d^1$, is the steepest descent step for the differentiable function $\lambda_1(x)$, projected to satisfy the linear constraints $Cd = 0$ and (38), and with steplength restricted by (37), (39) or (40) (if the last case applies, the algorithm terminates immediately with $d = \alpha^1 d^1$). When $t > 1$, the algorithm certainly does not yield a steepest descent direction; such a direction would violate (25) and hence split the current approximate multiple eigenvalue. However, the first m components of $\bar{d}^1$ may be viewed as a projected steepest descent direction, where by this we mean projected to satisfy the additional $t(t+1)/2-1$ conditions in (36).

The selection rule for α^0 in Step 0 eliminates one potential difficulty with the SQP method, namely the possibility of an infeasible subproblem.

Instead of using the PLP algorithm, which is based on QR factorizations of matrices with a small number of columns, an alternative approach would be to use an affine scaling interior point method to partially solve the LP.

We now define the successive partial linear programming (SPLP) method whose purpose is to solve the constrained model problem when m is large by a sequence of calls to the PLP algorithm. Each of these calls partially solves an LP of the form (35)–(40). The number of equality constraints in (36) is determined by the multiplicity estimate t. As with the SQP algorithm, the hope is that, once t is determined correctly, the inequality constraints (37) will become permanently inactive. However, since bounds in (11) may be active at a solution x^*, it is not adequate to begin the PLP algorithm with all bounds on the elements of d inactive, since then the same active set of bounds would have to be repeatedly built up every time the PLP algorithm is executed. This inefficiency is avoided by the use of the bound active set K. Bounds are added to K when they are encountered during a PLP execution; they are removed from K after a PLP execution if the corresponding multiplier signs indicate that they should not be active. Also, if the dual matrix U defined by the multipliers characterizing a PLP "solution" is indefinite, the multiplicity tolerance τ is reduced. The updating of the trust radius ρ is based on recommendations in [Fle87].

SPLP Algorithm to solve (10)–(11).

Step 0. Initialize the trust radius ρ and the multiplicity tolerance τ. Define a convergence tolerance ϵ. Set x to an initial value satisfying (11). Compute the eigenvalues and eigenvectors of $A(x)$. Initialize K to the empty set.

Step 1. Define the multiplicity estimate t and associated block of eigenvectors Q_1 by (22)–(23). Set $K^{'} = K$. Partially solve the LP (35)–(40), using the PLP Algorithm, producing $\bar{d} = (d^T, \delta)^T, v, \gamma$ and (a possibly modified) K.

Step 2. Construct U and μ from v, by setting diagonal elements of U to corresponding multipliers for diagonal equations of (25), off-diagonal elements of U to half the corresponding multipliers for the off-diagonal equations of (25), and elements of μ to corresponding multipliers for the constraint $Cd = 0$. If U is not positive semi-definite, reduce τ by a factor of two. If $\| d \| \leq \epsilon$, go to Step 5.

Step 3. Compute the eigenvalues of $A(x+d)$. If $\lambda_1(x+d) \geq \lambda_1(x)$, then set $K = K^{'}$, divide ρ by two and go to Step 1.

Step 4 Define

$$\psi = \frac{\lambda_1(x) - \lambda_1(x+d)}{-\delta},$$

the ratio of the actual to predicted reduction in the minimization objective. If $\psi > 0.75$, double ρ; if $\psi < 0.25$, divide ρ by two. Compute the eigenvectors of $A(x+d)$, if they were not already obtained, and replace x by $x+d$. If γ does not satisfy (32), remove indices from K corresponding to violated bounds in (32). Go to Step 1.

Step 5. If U is positive semi-definite and γ satisfies (32), stop. If U is not positive semi-definite then obtain a reduction in λ_1 by splitting the multiple eigenvalue $\lambda_1(x) = \cdots = \lambda_t(x)$, as explained in Theorem 7; then reduce τ by a factor of 10 and go to Step 1. Otherwise, if γ violates (32), remove indices from K corresponding to violated bounds in (32), divide ρ by two and go to Step 1.

The following theorem provides one justification for the SPLP method; to avoid unnecessary complication, some simplifying assumptions are made.

Theorem 12 *Suppose that the PLP algorithm called by Step 1 of the SPLP method generates $\bar{d} = (d^T, \delta)^T$ with the property that*

$$\bar{d} = \alpha^0 \bar{d}^0 + \alpha^1 \bar{d}^1,$$

i.e. no bound in (39) or constraint in (37) becomes active. Suppose also that U defined by the subsequent Step 2 of the SPLP method is positive semi-definite. Then d is a direction of non-ascent for λ_1.

Proof. By construction, we have $E\bar{d}^0 = e$, $E\bar{d}^1 = 0$, so $E\bar{d} = \alpha^0 e$. Therefore, by the same argument used in Theorem 10, (33) holds. We therefore wish to show that $\delta = \bar{d}_{m+1}$ is nonnegative. Let us first look at the second term of $\bar{d}$. We have

$$(\bar{d}^1)_{m+1} = \tilde{g}^T \tilde{d} = -\tilde{g}^T (I - YY^T)\tilde{g} \leq 0$$

where $\tilde{g}$, $\tilde{d}$ are defined by Step 1 of the PLP algorithm, since YY^T is the orthogonal projector onto the range space of G and $I - YY^T$ is the orthogonal projector onto the null space of G^T. Now consider the first term of $\bar{d}$. We have

$$Gv = YY^T\tilde{g}$$

so taking an inner product with (41) gives

$$v^T e = \tilde{g}^T Y d_Y = \tilde{g}^T \tilde{d}^0 = (\bar{d}^0)_{m+1}.$$

The proof is now complete, since $v^T e \leq 0$ for the same reason as given in Theorem 10. □

Theorem 12 is based to some extent on [MO80, Theorem 4]; as a point of comparison, note that the dual matrix estimate U generated by the SPLP method is obtained from least-squares approximation.

We expect that the algorithm described above will be modified in the future with further computational experience and theoretical development. In particular, we have no theoretical guarantee that the algorithm will converge to an optimal solution; we have not yet attempted any convergence analysis. However, in its present form, the algorithm has been used to obtain very satisfactory solutions to the problems to be described in Sections 9, 10 and 11.

Although it is not practical to compute W when m is large, we note that the SPLP method can probably be improved by approximating the second-order information in some way. The expression for W given by (4.12) of [Ove88] is actually a sum of terms, one corresponding to each eigenvalue smaller than λ_t. Since the denominator of each term is the separation of the eigenvalue from λ_1, one idea is to approximate W by a low rank approximation, consisting of terms corresponding to eigenvalues immediately lower than λ_t. It is not clear exactly how the low rank approximation would be exploited, but note that an SLP method may be regarded as an SQP method with a zero rank approximation to the quadratic term. An alternative idea is to approximate W using a limited memory quasi-Newton method; see [LN89]. In either case it seems probable that a practical SQP method could be devised which would converge faster than the SPLP method unless it had difficulty identifying the optimal multiplicity t^*.

We complete this section by noting that if n is large, the number of inequalities in (26), and therefore (37), should be substantially reduced. Indeed, as discussed in the next section, it is not practical to compute all the eigenvalues of $A(x)$ when n is large.

6 Computation of the Eigenvalues when n is Large

When n is large, the QR algorithm used by Eispack is not an efficient way to solve the eigenvalue problem. Indeed, it is particularly inappropriate for our purposes for two reasons:

1. Since only the largest eigenvalues are of any relevance to the optimization, it is grossly inefficient to compute all the eigenvalues of each matrix iterate $A(x)$.

2. Typically, each matrix iterate $A(x)$ generated by the optimization calculation does not differ much from the previous matrix iterate, whose eigenvalues and eigenvectors have already been computed.

For both of these reasons, it is clear that the eigenvalues should be computed by an iterative method. Possibilities are power methods, inverse power methods and Lanczos methods. The best choice depends on a number of considerations. In all cases, however, it is essential to iterate with a block of r vectors, which are approximate eigenvectors for $\lambda_1, \ldots, \lambda_r$, where $r \geq t^*$, the multiplicity of λ_1 at the optimal solution. Otherwise it will not be possible to verify the multiplicity t^* or to generate the dual matrix U. Indeed, unless an *a priori* upper bound on t^* is known, it is necessary that $r > t^*$ to be sure that the correct multiplicity is calculated. The number r can be adjusted during the iteration according to the value of the current multiplicity estimate t. It is important to maintain orthogonality of the r vectors during the iteration. The orthogonalized block versions of the power and inverse power methods are generally called subspace iteration; see Parlett [Par80] for details. The block of eigenvectors computed for the *previous* matrix iterate is a very valuable starting block for each subspace iteration after the first few optimization steps.

The simplest variation of subspace iteration is that based on the ordinary power method, which requires repeated multiplication of $A(x)$ onto the block of approximate eigenvectors. To be applicable, it is necessary that $\lambda_r(x) > |\lambda_n(x)|$; usually this method is used only when $A(x)$ is positive definite. The convergence of $\lambda_i(x), 1 \leq i \leq r$, depends on the separation of its magnitude from $\lambda_{r+1}(x)$. In particular, convergence of $\lambda_1(x)$ is fast if

$$\lambda_1(x) >> \lambda_{r+1}(x); \quad \lambda_1(x) >> |\lambda_n(x)| \,. \tag{44}$$

Whether or not (44) holds, a Lanczos method generally converges faster than the power method. However, a block Lanczos method is needed, for the reason just explained. We have not tried using block Lanczos since the necessary software has not advanced beyond an experimental stage.

Suppose now that $|\lambda_1(x)| < |\lambda_n(x)|$. This happens, in particular, if $A(x)$ is negative definite; equivalently, the optimization objective is to maximize the smallest eigenvalue of the positive definite matrix $-A(x)$, as in the column problem to be discussed in Section 9. In this case, an inverse block power method (subspace iteration) is appropriate. Convergence of $\lambda_1(x)$ is fast if

$$|\lambda_1(x)| << |\lambda_{r+1}(x)|; \quad |\lambda_1(x)| << |\lambda_n(x)| \,.$$

This is the case for the column problem. The inverse power method, unlike the power method, requires factorization of $A(x)$ at each step of the optimization

iteration, i.e. once per subspace iteration, as well as two triangular "solves" at each step of the subspace iteration.

If the power or inverse power methods converge slowly an attractive alternative is the shifted inverse power method, commonly known as inverse iteration. As before, the iteration must be carried out on a block of vectors. Each step requires the block of vectors to be multiplied by the inverse of $sI - A(x)$; this is implemented by a factorization of $sI - A(x)$ and several triangular "solves". An excellent shift s is available, namely the value of λ_1 from the previous matrix iterate. After the first few optimization steps, the shift is usually so good that only one shifted inverse multiplication is needed. If $sI - A(x)$ is discovered not to be positive definite during its factorization, the iterate x may be rejected immediately and the optimization trust radius ρ reduced, since $\lambda_1(x)$ is necessarily greater than the previous value s. This is a very valuable observation.

Whatever iterative method is chosen to compute the eigenvalues, caution must be used. In particular, if the iteration is terminated too soon with an inaccurate underestimate of λ_1, which is lower than the previous best value, the optimization algorithm may be unable to obtain a further reduction in $\lambda_1(x+d)$ for the simple reason that its estimate of $\lambda_1(x)$ is wrong. Thus a good implementation of the algorithm needs to allow recomputation of $\lambda_1(x)$ when necessary. We have not yet incorporated this automatically, instead restarting the algorithm when necessary. This is usually needed only at the beginning of the optimization if shifted inverse iteration is used, since the excellent choice of shift available makes this method very accurate. Note one fortunate fact: whatever form of block iteration is used, it is λ_1 which is the most accurately computed of $\lambda_1, \ldots, \lambda_r$; this is the eigenvalue whose accuracy is the most critical.

If factorizations are not practical, inverse or shifted inverse subspace iteration is still possible by the incorporation of a third nested iteration for, e.g., the conjugate gradient method to solve the linear systems required for each step of each subspace iteration. In the case of shifts, this inner iteration may be terminated if indefiniteness is detected, for the same reason explained above. We note however that the performance of the conjugate gradient method on the nearly singular systems that result from a good choice of shift is not very well understood. Most of our numerical experiments have used factorizations but some (not very extensive) experiments with a conjugate gradient version suggest that the method may give poor results when the shift is good, perhaps because of instability resulting from the near singularity of $sI - A(x)$. An alternative idea, following Szyld [Szy83], is to use the Paige and Saunders method SYMMLQ [PS78]. This may give better results than conjugate gradient for nearly singular positive definite systems. Szyld gives an argument explaining why the near singularity does not cause difficulty for SYMMLQ; he did not consider the conjugate gradient method, since he was concerned with interior eigenvalues and therefore needed to operate with indefinite systems. However, the disadvantage of using SYMMLQ is that the shifted inverse iteration may converge to a subdominant eigenvalue, since the iteration is not terminated when $sI - A(x)$

is indefinite. We have not yet experimented with a preconditioned conjugate gradient method, for example using a factorization of an earlier matrix iterate for a number of steps of the optimization.

7 The Generalized Eigenvalue Problem

All of the preceding sections may easily be generalized to apply to the eigenvalue problem

$$A(x)q = \lambda Bq \tag{45}$$

where B is a symmetric positive semi-definite matrix independent of x, not necessarily the identity matrix as has been implicitly assumed up to this point. We have the following modifications to Lemma 1 and Theorem 1 (proofs are omitted).

Lemma 2 *Let Q be a matrix $\in \Re^{n\times n}$ such that*

$$Q^T BQ = I. \tag{46}$$

Then the convex hull of the set

$$\{ww^T : w \in \Re^n, w^T Bw = 1\}$$

is the set

$$\{\tilde{U} = Q\hat{U}Q^T : \hat{U} \in \Re^{n\times n},\ \hat{U} = \hat{U}^T,\ \text{tr}\ \hat{U} = 1,\ \hat{U} \geq 0\}.$$

Furthermore, the elements in the first set are the extreme points of the second set.

Note that the trace of $\tilde{U}$ is generally not equal to one.

Theorem 13 *As above, let Q be any matrix $\in \Re^{n\times n}$ satisfying $Q^T BQ = I$. Now let $\lambda_1(A, B)$ denote the largest eigenvalue of the pencil (A, B), i.e. largest root λ of (45) for nontrivial q, ignoring for the moment the dependence of A on x. The following characterizations of λ_1 hold:*

$$\lambda_1(A, B) = \max\{\langle q, Aq\rangle \ : q^T Bq = 1\};$$

$$\lambda_1(A, B) = \max\{\langle qq^T, A\rangle \ : \ q^T Bq = 1\};$$

$$\lambda_1(A, B) = \max\{\langle \tilde{U}, A\rangle \ : \ \tilde{U} = Q\hat{U}Q^T,\ \hat{U} \in S\Re^{n\times n},\ \text{tr}\ \hat{U} = 1,\ \hat{U} \geq 0\}. \tag{47}$$

Now take $Q = [q_1, \ldots q_n]$ to be a matrix of eigenvectors of (A, B), normalized so that (46) holds. Thus, in addition to (46), we have

$$Q^T AQ = \text{Diag}(\lambda_i).$$

Assume that the largest eigenvalue λ_1 has multiplicity t, with corresponding eigenvectors, $q_1, \ldots, q_t$ making up a matrix $Q_1 \in \Re^{n \times t}$. We see then that the set of matrices achieving the max in (47) is, as before, the right-hand side of (6). Indeed, Theorem 2 and all subsequent theorems, remarks and algorithm statements then apply exactly as before provided only that *the normalization (46) is consistently used for the eigenvectors.*

Note that the details of subspace iteration are well known for the generalized problem; see [BW76],[Par80, Ch. 15]. If a shift s is used, it is of course understood that $A(x)$ is to be shifted by sB instead of sI.

8 Several Matrix Families

Suppose it is desired to minimize

$$\phi(x) = \max_{1 \le l \le p} \lambda_1^{(l)}(x), \tag{48}$$

subject to (11), where each $\lambda_1^{(l)}(x), l = 1, \ldots, p$, is the largest eigenvalue of a matrix-valued function $A^{(l)}(x)$. The necessary optimality conditions are easily extended to this case by introducing a dual matrix for each matrix family. Given x, let t_l be the multiplicity of $\lambda_1^{(l)}(x)$ if the latter quantity equals $\phi(x)$, and zero otherwise. Let $Q_1^{(l)}$ be an orthonormal set of t_l corresponding eigenvectors if $t_l > 0$, and the empty matrix otherwise.

Theorem 14 *A necessary condition for x to solve (48), (11) is, in addition to (11), that there exist dual matrices $U^{(l)} \in S\Re^{t_l \times t_l}$, $l = 1, \ldots, p$, and vectors of Lagrange multipliers $\mu \in \Re^{n_c}$ and $\gamma \in \Re^m$, satisfying*

$$\sum_{l=1}^{p} \langle U^{(l)}, (Q_1^{(l)})^T A_k(x) Q_1^{(l)} \rangle \;=\; \langle \mu, c_k \rangle \;+\; \gamma_k, \quad k = 1, \ldots, m \tag{49}$$

$$\sum_{l=1}^{p} \mathrm{tr}\; U^{(l)} = 1 \tag{50}$$

$$U^{(l)} \ge 0, \quad l = 1, \ldots, p, \tag{51}$$

as well as (15). The necessary condition is also sufficient in the affine case.

The proof is a straightforward generalization of the proof of Theorem 5.

Similarly, the SQP and SPLP algorithms are easily adapted to minimize $\phi(x)$ by including in the QP or LP constraints of the form (25)–(26) for each of the p matrix families. Multiplicity estimates $t_l, l = 1, \ldots, p$, may be defined as the largest integer t such that

$$\phi(x) - \lambda_t^{(l)}(x) \le \tau \max(1, |\,\phi(x)\,|),$$

with $t_l = 0$ if no positive integer satisfies the inequality. Note that it is *not* recommended to simply define $A(x)$ to be a block diagonal matrix with blocks $A^{(l)}(x), l = 1, \ldots, p$. Such an approach loses some of the structure of the generalized gradient of $\phi(x)$, since it does not take account of the fact that eigenvalues corresponding to different diagonal blocks of a block diagonal matrix do not interact with each other.

One application of (48) is minimizing the maximum eigenvalue of $A(x)$ in absolute value by taking $A^{(1)}(x) = A(x), A^{(2)}(x) = -A(x)$; see [Ove88] as well as Section 10 below.

9 The Column Problem

A classical problem which goes back to Lagrange is to find the shape of the strongest column with given volume. Mathematically the problem is to determine a function $\sigma(x)$, the cross-sectional area of the column, from an admissible set

$$\sigma \in L^\infty : 0 < \alpha \leq \sigma(x) \leq \beta, \int_0^1 \sigma(x)dx = 1 \tag{52}$$

to maximize the least eigenvalue of

$$-(\sigma^p(x)u''(x))'' = \lambda u''(x), \quad u \in H_0^2, \tag{53}$$

on the interval $[0, 1]$, where $p \geq 1$ (usually $p = 1$ or $p = 2$). Here p has a different meaning from the previous section and x refers not to unknown parameters but to a spatial dimension along the axis of the column. The function $u(x)$ measures the displacement of the column when deflected from its equilibrium position. The case $p = 2$ models columns with circular (or equivalently square) cross-sections of uniform material. The case $p = 1$ models "thin-wall" beams or columns, where a variable thickness shell of one kind of material surrounds a uniform core of another material. The significance of the least eigenvalue of the differential equation is that it corresponds to the critical buckling load in the Euler-Bernoulli model of the column. (The load is applied at the ends of the column, in the direction of its axis.)

The problem is a controversial one which has been addressed by many applied mathematicians and structural engineers, including [TK62], [OR77]. Our work on this problem is joint with Steve Cox; the details of our theoretical and computational contributions may be found in [CO91]. Here we briefly summarize some of the computational results. We discretized the problem, approximating $\sigma(x)$ by a piecewise constant function σ_h, where h is the mesh size. Following the standard approach in [SF73], we approximated u by u_h, using piecewise cubic Hermite finite elements, and constructed the corresponding finite-dimensional bending matrix $A(\sigma_h)$ and stiffness matrix B such that the eigenvalues of the generalized problem (45)

$$A(\sigma_h)q = \lambda Bq \tag{54}$$

converge to the eigenvalues of the differential equation as h decreases to zero. (Only the smallest eigenvalues are well approximated by the discretization; these are also the eigenvalues of physical interest.) The eigenvector q consists of the values of u_h and its derivative at the mesh points. There is a slight conflict of notation; σ_h refers both to a piecewise constant function and to the vector of variables which defines it. The boundary conditions of (53) are "clamped-clamped"; thus A and B are defined so that u_h and its derivative are zero at 0 and 1. Note that, as in (45), A depends on the unknown variables while B does not. The integral constraint in (52) becomes a linear constraint on σ_h. Regarding the bounds on σ: a solution of the mathematical problem is known to exist only for $\alpha > 0$, $\beta < \infty$ [CO91]; however, in practice these requirements do not seem to be necessary and for most experiments we used $\alpha = 0$, $\beta = \infty$.

We then applied the SPLP algorithm of Section 4 to find that σ_h which maximizes the smallest eigenvalue of (54), or equivalently, negating the signs of the eigenvalues, minimizes the largest one, subject to the linear integral constraint. The order of the matrices A and B, n, is $2N-4$, and the number of variables, m, is $N-1$, where $N = h^{-1}+1$. We used the inverse version of subspace iteration without shifts to compute the eigenvalues, which requires the factorization of a band matrix at each optimization step, as explained in Section 6. Since it is known that the extremal eigenvalue cannot have multiplicity greater than 2, we computed only $r = 2$ eigenvalues. Most of the papers in the literature do not take this direct optimization approach. Of the few that do, we do not know of any that computes the dual matrix approximation U, which is the key to verifying optimality. (When $p = 1$, $A(\sigma_h)$ is linear, so the minimum eigenvalue is concave; when $p > 1$, concavity is lost, and satisfaction of the necessary conditions does not guarantee optimality, but comparison of the results for varying p indicates that our computed solutions are most likely global maxima.)

The results show that when $p > 1$, the optimal σ_h is bounded away from zero as $h \to 0$, but for $p = 1$ apparently the optimal σ_h converges to zero at two points as $h \to 0$. Presumably, the optimal column has zero thickness at two points if $p = 1$, but not if $p > 1$. This has been a subject of great controversy in the literature, especially when $p = 2$; see [CO91] for details. Plots of the optimal cross-sectional area $\sigma_h(x)$ are shown in Figure 1 for $N = 513$ with $p = 1$ and $p = 2$ respectively. The functions plotted are piecewise constant with 512 pieces, with no interpolation. The strongest column is about 33% stronger than the uniform column with the same volume in the case $p = 2$ and about 25% stronger in the case $p = 1$.

In all cases $1 \le p \le 3$, the first eigenvalue is double at optimality. It is this double eigenvalue which has caused most of the debate in the literature; indeed, some authors have expressed doubt about the multiplicity even when giving the correct result for the optimal σ. Even more interesting, the 2×2 dual matrix U which demonstrates optimality has minimum eigenvalue bounded away from zero as $h \to 0$ for all $p > 1$, but for $p = 1$ the dual matrix is apparently singular in the limit as $h \to 0$. We conclude that the double multiplicity of the eigenvalue

of the differential equation is "strongly stable" for $p > 1$, but not for $p = 1$.

The performance of the SPLP algorithm was very good. The results shown in Figure 1 were obtained using a convergence tolerance $\epsilon = 10^{-3}$, with the multiplicity tolerance and trust radius set initially to $\tau = 0.1$ and $\rho = 5$ and the variables initialized to 1, i.e. the uniform column. The number of calls to the subspace iteration routine, i.e. the number of times the computation of the eigenvalues was required, was 60 for $p = 2$ and 27 for p=1, with a total computation time of less than 1.5 hours on a Sparc station in each case. The residual of equations (12)–(13) was reduced to about 10^{-3} in the case $p = 2$ and about 10^{-2} in the case $p = 1$. The accuracy of the optimal λ_1 was approximately 4 decimal figures, with the gap between the first and second eigenvalues reduced to about 10^{-6}. Such fast convergence indicates a well conditioned optimization problem, since the method is only first-order. We also performed some experiments with α, the lower bound on σ_h, set to a positive number e.g. 0.25. The active bound strategy used by the SPLP algorithm worked very effectively. Typically most of the active bounds were identified in just a few steps, with fine tuning of the active set taking place subsequently.

10 Design of Optimal Preconditioners

Greenbaum and Rodrigue [GR90] have used our optimization programs to solve the following problem: given a positive definite symmetric matrix B, find the positive definite symmetric matrix M with prescribed sparsity pattern which minimizes the 2-norm condition number of $M^{-1/2}BM^{-1/2}$. They show that M equivalently minimizes the maximum eigenvalue (in absolute value) of

$$I - M^{-1/2}BM^{-1/2}$$

or

$$I - L^{-1}ML^{-T}, \tag{55}$$

where LL^T is a Cholesky factorization of B. The latter formulation is preferable, since the variables, namely the nonzero elements of the sparse matrix M, enter linearly. Since a factorization of B is used, finding the optimal preconditioner is clearly much more costly than solving a system $Bx = b$; the idea is that finding such optimal preconditioners gives insight which can then be widely applied.

The work reported in [GR90] was done before the SPLP version of the algorithm was available, so the SQP method described in Section 3 was used, the eigenvalues being computed by Eispack. The primary interest was in matrices B arising from elliptic partial differential equations, but only very coarse meshes could be handled. Nonetheless, it was found that the experiments gave a substantial amount of insight. For example, the optimal tridiagonal preconditioner M for B equal to the five-point finite difference approximation to the Laplacian on the square was computed. The results led to the conjecture that the

optimal condition number is $O(h^{-2})$, where N is the number of mesh points in each direction, and that the optimal tridiagonal preconditioner is only slightly better than simply setting M to be the tridiagonal part of B. It was also found that the optimal solution yields (55) with a double eigenvalue at each end of its spectrum, these two double eigenvalues having the same magnitude. Further experiments involving domain decomposition were also done; this is a promising area for further investigation.

A better way to formulate the optimization problem is to minimize the maximum eigenvalue, in absolute value, of the generalized eigenvalue problem

$$(M - B)q = \lambda Bq.$$

Note that, as in Section 7, the variables, namely the elements of M, appear only on the left-hand side. Using this formulation, we have now performed further experiments with the SPLP version of our algorithm. Our first idea was to compute the extreme eigenvalues of the pencil $(M - B, B)$ by direct subspace iteration. This requires only one Cholesky factorization of B before the optimization iteration begins. However, convergence was much too slow for this approach to be practical. We therefore used shifted inverse iteration to independently compute the algebraically largest eigenvalues of the pencils

$$(A^{(1)}, B) = (M - B, B) \text{ and } (A^{(2)}, B) = (B - M, B).$$

This required factorizations of $(s + 1)B - M$ and $(s - 1)B + M$ at each optimization step, for which we used the Linpack band matrix subroutines. At the optimal solution of all test problems, and indeed usually after a few optimization steps, the largest and smallest eigenvalues of $(M - B, B)$ were approximately equal in magnitude and opposite in sign. As explained in Section 8, two dual matrices $U^{(1)}$ and $U^{(2)}$ are generated by the SPLP algorithm, with dimensions t_1 and t_2 which are the computed multiplicities of each end of the spectrum of $(M - B, B)$. Note that instead of (13), we have the condition

$$\text{tr } U^{(1)} + \text{tr } U^{(2)} = 1.$$

We computed the optimal banded preconditioner M for B equal to the finite difference negative Laplacian on a unit square with mesh size h in each direction. We assumed Dirichlet boundary conditions, so that B and M are $n \times n$ matrices, where $n = N^2$, $N = h^{-1} - 1$. The matrix M is said to have half-bandwidth k if its total bandwidth is $2k+1$; thus, for $k = 0$, M is restricted to be diagonal, while if $k = N$ the optimal solution is $M = B$. The dimension of the optimization problem, m, is approximately $(k + 1)N^2$. The results support the following conjecture: the optimal preconditioner M with half-bandwidth k gives a pencil $(M - B, B)$ with eigenvalues of multiplicity $k+1$ at each end of its spectrum, for all $k < N$. However, computing accurate optimal preconditioners for even moderate mesh sizes was very difficult for the simple reason that, like

the discrete Laplacian itself, the eigenvalue optimization problem is increasingly ill-conditioned as N increases. The negative end of the spectrum of $(M-B, B)$ has a cluster of eigenvalues which becomes more dense as N increases. For small mesh sizes ($N \leq 6$) there was not much difficulty identifying the apparently correct optimal multiplicity $k+1$, but this became more difficult for larger N, since the gap between the extremal eigenvalue and the interior eigenvalues becomes smaller as N increases. Furthermore, it is apparently the case that $\mathrm{tr}\, U^{(1)} \to 0$ and $\mathrm{tr}\, U^{(2)} \to 1$ as $N \to \infty$, showing that the positive end of the spectrum of $(M-B, B)$ becomes more and more irrelevant as the discrete Laplacian B becomes closer to being singular.

The situation is quite different from that reported for the column problem as we allow the mesh size to go to zero. The column problem is well posed in infinite dimensions and the finite dimensional optimization problem is well conditioned as $N \to \infty$. By contrast, the optimal preconditioning problem for the Laplacian is not a well posed problem in infinite dimensions. The reason for this is that the column problem is concerned only with one end of the spectrum of the differential operator, namely the lowest eigenvalue which corresponds (in the case that it is simple) to a positive eigenfunction, while the optimal preconditioning problem is concerned with both ends of the spectrum, including eigenvalues corresponding to highly oscillatory eigenfunctions.

The computed optimal spectral radius of $(M-B, B)$ is plotted in Figure 2 for various k and N. The trend is clear. The optimal tridiagonal preconditioner represents a significant improvement over the optimal diagonal preconditioner (which is a scalar multiple of the identity matrix). However, increasing k gives successively smaller improvements until k starts to approach N. This, of course, reflects the fact that the discrete Laplacian has only five nonzero diagonals, namely the three main diagonals and the Nth sub- and super-diagonal.

11 A Graph Problem

The following problem was communicated to us by H. Schramm and J. Zowe; its origin may be found in [Lov79], [GLS88]. Given an undirected graph G, with vertices $1, \ldots, n$, let M be an $n \times n$ symmetric matrix with the restriction that its diagonal elements are zero and its offdiagonal elements (i, j) are zero if i and j are not adjacent in the graph, and let x be the vector whose components are the nonrestricted lower triangular elements of M. The problem is to choose M, or equivalently x, to minimize the largest eigenvalue of

$$A(x) = M + ee^T \tag{56}$$

where $e = [1, \ldots, 1]^T$. The minimum value for the max eigenvalue is known to give an upper bound for the Shannon capacity of the graph [Lov79]. (The upper bound is sometimes called the Lovasz number of the graph.)

We applied our eigenvalue optimization algorithm to a test problem suggested by [Sch89a]. Given integers $\alpha \geq 1$ and $\omega \geq 3$, let $n = \alpha\omega + 1$ and define G to have the property that vertices i and j are adjacent if $j - i < \omega$ or $i + n - j < \omega$. The class of graphs with this property is denoted $C_n^{\omega-1}$. We tried solving the optimization problem for for various values $\alpha \leq 10$ and $\omega \leq 6$. For these examples the order of the matrix, n, is moderate (≤ 61), but the number of variables, m, which is the number of pairs of adjacent vertices in the graph, is large (≤ 305). Consequently it is important to use the SPLP version of the optimization algorithm, but it is reasonable (though not very efficient) to compute the eigenvalues using Eispack. (Unshifted subspace iteration would not work since the smallest eigenvalue, which is of no interest, is negative and sometimes has a larger magnitude than the largest eigenvalue.)

The test problems are certainly very interesting. In all cases the algorithm *immediately* generated a point, say $\hat{x}$, where the max eigenvalue is multiple to machine precision, with the two optimality conditions (12)–(13) satisfied to machine precision. The multiplicity was seven in the cases where $\omega = 4$ and eleven in the cases where $\omega = 6$. (In some cases this required as many as four optimization steps, since successive doubling of the trust radius was needed to make a sufficiently large change in x.) In the case of the first two test problems, the dual matrix U was positive semi-definite and the algorithm terminated with the optimal solution $\hat{x}$. In all other cases, however, the dual matrix U was not positive semi-definite and so it was necessary for the algorithm to split the multiple eigenvalue to obtain a lower point, as described in Theorem 7. The algorithm then took many more steps to converge to the optimal solution x^*. In all these cases, the max eigenvalue had the same multiplicity at the final solution x^* as at the initially generated point $\hat{x}$. This unusual behavior of the algorithm indicates some underlying linear structure of the eigenvalues which is not generic and not well understood at the present.

In general, it seems that the optimal multiplicity is $2\omega - 1$. Another interesting observation is that the minimum eigenvalue of the optimal dual matrix has multiplicity two for all the problems we have run.

The results are summarized in Table 1. The first two columns specify the problem, and the third gives the number of variables. The next three columns give the computed optimal max eigenvalue, its multiplicity, and the smallest eigenvalue of the associated dual matrix U. The last value given is the number of times the eigenvalues of $A(x)$ were computed (using Eispack). The convergence tolerance was set to $\epsilon = 10^{-6}$. The multiplicity tolerance and trust region radius were initialized to $\tau = .01$ and $\rho = 10$ respectively. The variables were all initialized to -1. The norm of the residual of equations (12)–(13) was reduced in each case to about 10^{-6}, except in the first two cases where it was reduced to machine precision (about 10^{-14}) in one step. In the two cases marked by an asterisk (*) it was necessary to restart the algorithm at one point (with the original values of τ and ρ) to obtain a satisfactory residual for (12)–(13). It is not clear why the case $\alpha = 8$, $\omega = 6$ was so much more difficult than the others,

α	ω	m	λ_1	t	min e.v.(U)	# λ-evals.
3	4	39	3.106027	7	.0532	1
4	4	51	4.132934	7	.0545	1
5	4	67	5.151476	7	.0556	217
8	4	99	8.183308	7	.0575	130
10	4	123	10.195149	7	.0584	219
3	6	95	3.055559	11	.0195	235
4	6	125	4.073890	11	.0209	238
5	6	155	5.087257	11	.0219	187
6	6	185	6.097343	11	.0227	181
7	6	215	7.105194	11	.0233	227
8	6	245	8.111465	11	.0237	957*
9	6	275	9.116589	11	.0241	478
10	6	305	10.120845	11	.0244	608*

Table 1: Summary of Results for Graph Problem

but in all cases an accurate solution was found eventually.

It is of some interest to compare our algorithm to that used by Schramm and Zowe, a "bundle trust region" method, which, as the name suggests, combines ideas of trust region methods with those of the early subgradient bundle methods of Lemarechal [LM78]. This algorithm is intended for general nonsmooth optimization problems, not necessarily involving eigenvalues. The bundle trust region method accumulates a set ("bundle") of subgradients during the course of the optimization. In the version described in [Sch89b,Zow89], one subgradient is added to the bundle per iteration, namely

$$[q^T A_1(x)q, \ldots, q^T A_m(x)q]^T, \tag{57}$$

where, as earlier, $A_k(x) = \frac{\partial A(x)}{\partial x_k}$ (in this case a matrix with one nonzero element), and where q is a normalized eigenvector corresponding to $\lambda_1(x)$, arbitrarily chosen from the invariant subspace if the multiplicity of $\lambda_1(x)$ is greater than one. Theorem 2 (together with the chain rule) assures us that this vector is indeed a subgradient of $\lambda_1(x)$, that is, an element of the generalized gradient $\partial\lambda_1(x)$.

The initial comparison of our results with those of Schramm and Zowe showed that, while both algorithms obtained accurate solutions, our algorithm usually required fewer steps to achieve the same accuracy [Sch89a]. However, a revised version of Schramm and Zowe's algorithm has now been tested, where at each iteration, if $\lambda_1(x)$ has approximate multiplicity t, then t subgradients of the form (57) are added to the bundle of subgradients, for q equal to the t different columns of the matrix of eigenvectors $Q_1(x)$. This strategy substantially im-

proved the algorithm which now requires far fewer steps than ours for the same accuracy [Sch89a]. The reason for the dramatic improvement is not completely clear, but it may be related to the surprising initial behavior of our algorithm. Considering (6) in Theorem 2 again, we see that the first version of Schramm and Zowe's algorithm computes the subgradient defined by $U = e_1 e_1^T$, while the second version computes the t subgradients defined by $U = e_k e_k^T, k = 1, \ldots, t$ (here e_k is the kth column of the identity matrix). Clearly, then, one could add more subgradients to the bundle, using other permissible values for U; there is nothing special about the choice $U = e_k e_k^T$, since the basis Q_1 has been arbitrarily chosen by Eispack. The feature of our algorithm which we believe to be very attractive is that it efficiently computes $t(t+1)/2$ generically linearly independent subgradients at each iteration, namely the gradients of the structure functionals (8), while the dual matrix estimate U defines the linear combination of these subgradients which satisfies the optimality condition (12) in the limit. This dual matrix is the key not only to the verification of optimality but also to any sensitivity analysis of the solution (see Theorem 7).

It would be premature to draw conclusions as to whether the bundle trust region algorithm or ours is more efficient, for several reasons: the former requires an estimate of the optimal solution value, which ours does not; the former solves a QP (with dimension equal to the number of subgradients in the bundle), which ours does not; comparisons have been made only on the graph problems just described, which apparently have a rather special structure which is not completely understood. We expect that it should be possible to improve the rate of convergence of our algorithm by approximating second-order information (see Section 5). We also wonder if the bundle trust region algorithm would have difficulties when the eigenvalues are computed by a shifted iterative method, since the basis Q_1 would tend to be little changed at each iteration. By contrast, when Eispack is used the basis $Q_1(x)$ does not generally converge as $x \to x^*$ (see the examples in [FNO86]), perhaps giving a bundle which is more "rich" in the various possible values for the subgradients.

Finally, we note that the dual matrix itself appears in the references [Lov79], [GLS88]. Indeed, the property stated as Theorem 4 in [Lov79] and the third equality in Theorem 9.3.12 of [GLS88] is a special case of Theorem 6 given above, specifically giving the dual formulation (16). It seems likely that the multiplicity of the minimum eigenvalue of the optimal dual matrix U (found to be two in our experiments), as well as the multiplicity of the optimal maximum eigenvalue of $A(x)$ (conjectured to be 2ω), should be significant for the understanding of the original graph capacity problem.

12 Concluding Remarks

We have derived optimality conditions for an important eigenvalue optimization model problem, emphasizing the representation of the generalized gradient in

terms of a dual matrix U. We have given a practical algorithm for solving large-scale problems of this type, based on successive partial linear programming, which has been applied very successfully in diverse application areas. The behavior of the algorithm was quite different for the three applications described in detail. The column problem described in Section 9 is a well-posed infinite dimensional optimization problem; discretized versions were solved very efficiently by the algorithm. The preconditioning problem described in Section 10 gave rise to very ill-conditioned problems which were nonetheless solved by the algorithm to reasonable accuracy. The algorithm also gave very accurate answers to the graph problems described in Section 11, which have a rather special structure which is not completely understood.

The SQP algorithm of [Ove88], on which the new algorithm is based, has also been applied to some other applications not discussed in this paper, including the quadratic assignment problem [RW91], the stability of Runge-Kutta methods for ordinary differential equations [Mul90] and optimal diagonal scaling of nonsymmetric matrices [Wat91]. Another application to which we hope to apply our large-scale algorithm is the computation of structured singular values in control [FT86,Wat90].

Perhaps the most important feature of our algorithms is that they compute the optimal dual matrix U, which is the key to the verification of optimality and to sensitivity analysis of the solution. Given the optimal dual eigenspace basis Q_1^*, the dual matrix U is unique if the active linear constraints of the limiting LP or QP are independent (see Theorem 9). If the linear independence assumption fails to hold, the problem is said to be degenerate, since U is then not uniquely defined and verification of optimality is much more difficult; this happens, for example, in the Runge-Kutta problems of [Mul90]. Because the basis Q_1^* may be replaced by any other orthonormal basis spanning the same eigenspace, it is the eigenvalues of U which are of significance. Nonnegativity of the eigenvalues of U is a necessary condition for optimality and, together with the other conditions of Theorem 5, a sufficient condition if $A(x)$ is affine. The eigenvalues of U play essentially the same role in sensitivity analysis of optimal solutions as that well known for dual variables (Lagrange multipliers) in the context of nonlinear programming; see Theorem 7. In particular, if the smallest eigenvalue of U is zero, it may be concluded that the optimal multiplicity of the minimization objective $\lambda_1(x)$ is not strongly stable.

Acknowledgements. The work on the column problem is joint with Steve Cox; much of the development of the large-scale versions of the algorithm described in Sections 5 and 6 was also joint work with Cox in the course of obtaining solutions to the column problem. The work on optimal preconditioners is joint with Anne Greenbaum. The presentation of the optimality conditions in Theorems 1 – 3 is related to [OWa], which is joint work with Rob Womersley. I would like to thank Helga Schramm and Jochem Zowe for providing me with details of the graph problems and the related performance of their bundle trust region method. I have also received much helpful input from many other

people, too numerous to list here, which I nonetheless gratefully acknowledge. This work was supported in part by National Science Foundation Grant No. CCR-88-02408.

References

[All89] J.C. Allwright. On maximizing the minimum eigenvalue of a linear combination of symmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 10:347–382, 1989.

[Boy87] S.P. Boyd, 1987. Private communication.

[Bra86] A. Bratus. Multiple eigenvalues in problems of optimizing the spectral properties of systems with a finite number of degrees of freedom. *USSR Journal on Computational Mathematics and Mathematical Physics*, 26:1–7, 1986.

[BW76] K. Bathe and E. Wilson. *Numerical methods in finite element analysis*. Prentice Hall, Englewood Cliffs, N.J., 1976.

[CDW75] J. Cullum, W.E. Donath, and P. Wolfe. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. *Mathematical Programming Study*, 3:35–55, 1975.

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley, New York, 1983.

[CO91] S.J. Cox and M.L. Overton. The optimal design of columns against buckling. *SIAM Journal on Mathematical Analysis*, 1991. To appear.

[D+78] J.J. Dongarra et al. *LINPACK Users Guide*. Society of Industrial and Applied Mathematics, Philadelphia, 1978.

[Doy82] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEEE Proc.*, 129 Pt. D:242–250, 1982.

[Ehr79] R.M. Ehrdahl. Two algorithms for the lower bound method of reduced density matrix theory. *Reports on Mathematical Physics*, 15:147–162, 1979.

[Fle85] R. Fletcher. Semi-definite constraints in optimization. *SIAM Journal on Control and Optimization*, 23:493–513, 1985.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley, Chichester and New York, second edition, 1987.

[FNO86] S. Friedland, J. Nocedal, and M.L. Overton. Four quadratically convergent methods for solving inverse eigenvalue problems. In D.F. Griffiths, editor, *Numerical Analysis*, pages 47–65, New York, 1986. John Wiley. Pitman Research Note in Mathematics 140.

[FNO87] S. Friedland, J. Nocedal, and M.L. Overton. The formulation and analysis of numerical methods for inverse eigenvalue problems. *SIAM Journal on Numerical Analysis*, 24:634–667, 1987.

[FT86] M.K.H. Fan and A.L. Tits. Characterization and efficient computation of the structured singular value. *IEEE Transactions on Automatic Control*, 31:734–743, 1986.

[GLS88] M. Grötschel, L. Lovász, and A. Schriver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, New York, 1988.

[GMSW86] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. User's guide for LSSOL: A Fortran package for constrained linear least-squares and convex quadratic programming. Systems Optimization Laboratory Report 86-1, Stanford University, 1986.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, New York and London, 1981.

[GO89] C.B. Gurwitz and M. L. Overton. Sequential quadratic programming methods based on approximating a projected Hessian matrix. *SIAM Journal on Scientific and Statistical Computing*, 10:631–653, 1989.

[Gol87] B. Gollan. Eigenvalue perturbations and nonlinear parametric optimization. *Mathematical Programming Study*, 30:67–81, 1987.

[GR90] A. Greenbaum and G.H. Rodrigue. Optimal preconditioners of a given sparsity pattern. *BIT*, 29:610–634, 1990.

[GT88] C.J. Goh and K.L. Teo. On minimax eigenvalue problems via constrained optimization. *Journal of Optimization Theory and Applications*, 57:59–68, 1988.

[GV83] G.H. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1983.

[IT79] A.D. Ioffe and V.M. Tihomirov. *Theory of Extremal Problems*. North-Holland, Amsterdam, 1979.

[Kat82] T. Kato. *A Short Introduction to Perturbation Theory for Linear Operators*. Springer-Verlag, New York, 1982.

[Lan64] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numerische Mathematik*, 6:377–387, 1964.

[LM78] C. Lemarechal and R. Mifflin, editors. *Nonsmooth Optimization.* Permagon Press, Oxford, 1978.

[LN89] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[Lov79] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[MO80] W. Murray and M. L. Overton. A projected Lagrangian algorithm for nonlinear minimax optimization. *SIAM Journal on Scientific and Statistical Computing*, 1:345–370, 1980.

[Mul90] M. Muller. *Algebraische Stabilitätsbedingungen für Runge-Kutta-Verfahren.* PhD thesis, Universität Karlsruhe, 1990.

[NAG] NAG library manual. Numerical Algorithms Group, Oxford.

[OR77] N. Olhoff and S. Rasmussen. On single and bimodal optimum buckling loads of clamped columns. *Int. J. Solids Struct.*, 9:605–614, 1977.

[Osb85] M.R. Osborne. *Finite Algorithms in Optimization and Data Analysis.* John Wiley, Chichester and New York, 1985.

[OT83] N. Olhoff and J.E. Taylor. On structural optimization. *Journal of Applied Mechanics*, 50:1138–1151, 1983.

[Ove88] M.L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 9:256–268, 1988.

[OWa] M.L. Overton and R.S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. Work in progress.

[OWb] M.L. Overton and R.S. Womersley. Second derivatives for optimizing eigenvalues of symmetric matrices. Work in progress.

[OW88] M.L. Overton and R.S. Womersley. On minimizing the spectral radius of a nonsymmetric matrix function – optimality conditions and duality theory. *SIAM Journal on Matrix Analysis and Applications*, 9:473–498, 1988.

[Pan89] E.R. Panier. On the need for special purpose algorithms for minimax eigenvalue problems. Technical report, Dept. of Elec. Eng., University of Maryland, 1989.

[Par80] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice Hall, Englewood Cliffs, N.J., 1980.

[PS78] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12:617–629, 1978.

[PW83] E. Polak and Y. Wardi. A nondifferentiable optimization algorithm for structural problems with eigenvalue inequality constraints. *Journal of Structural Mechanics*, 11:561–577, 1983.

[Roc70] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, N.J., 1970.

[RW91] F. Rendl and H. Wolkowicz. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming*, 1991. To appear.

[S+67] B.T. Smith et al. *Matrix Eigensystem Routines – EISPACK Guide*. Lecture Notes in Computer Science 6. Springer-Verlag, New York, 1967.

[Sch89a] H. Schramm, 1989. Private communication.

[Sch89b] H. Schramm. *Eine Kombination von Bundle- und Trust-Region-Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*. PhD thesis, Universität Bayreuth, 1989.

[SF73] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice Hall, Englewood Cliffs, N.J., 1973.

[Sha85a] A. Shapiro. Extremal problems on the set of nonnegative definite matrices. *Linear Algebra and its Applications*, 67:7–18, 1985.

[Sha85b] A. Shapiro. Optimal block diagonal l_2-scaling of matrices. *SIAM Journal on Numerical Analysis*, 22:81–94, 1985.

[Szy83] D.B. Szyld. *A two-level iterative method for large sparse generalized eigenvalue calculations*. PhD thesis, Dept. of Math., New York University, 1983.

[TK62] I. Tadjbakhsh and J.B. Keller. Strongest columns and isoperimetric inequalities for eigenvalues. *Journal of Applied Mechanics*, 29:159–164, 1962.

[Wat90] G.A. Watson. Computing the structured singular value, and related problems. In D.F. Griffiths, editor, *Numerical Analysis 1989*, pages 258–275, New York, 1990. John Wiley. Pitman Research Notes in Mathematics 228.

[Wat91] G.A. Watson. An algorithm for optimal l_2 scaling of matrices. *IMA Journal on Numerical Analysis*, 1991. To appear.

[Zho88] Jian Zhou, 1988. Private communication.

[Zow89] J. Zowe. The BT-algorithm for minimizing a nonsmooth functional subject to linear constraints. In F.H. Clarke, V.F. Demyanov, and F. Gianessi, editors, *Nonsmooth Optimization and Related Topics*, pages 459–480, New York, 1989. Plenum Press.