SIAM J. OPTIM.
Vol. 19, No. 1, pp. 281–306

A TRUST REGION SPECTRAL BUNDLE METHOD FOR NONCONVEX EIGENVALUE OPTIMIZATION*

P. APKARIAN†, D. NOLL‡, AND O. PROT‡

Abstract. We present a nonsmooth optimization technique for nonconvex maximum eigenvalue functions and for nonsmooth functions which are infinite maxima of eigenvalue functions. We prove global convergence of our method in the sense that for an arbitrary starting point, every accumulation point of the sequence of iterates is critical. The method is tested on several problems in feedback control synthesis.

Key words. eigenvalue optimization, trust region method, proximity control, spectral bundle, output feedback control, H_∞-synthesis

AMS subject classifications. 90C26, 90C22, 93B36, 49J52

DOI. 10.1137/060665191

1. Introduction.

Eigenvalue optimization has a wide spectrum of applications in physics, engineering, statistics, and finance. This spectrum includes composite materials [15], quantum computational chemistry [60], optimal system design [47, 8], shape optimization [17], pole placement in linear system theory, robotics [44], relaxations of combinatorial optimization problems [26, 39], experimental design [55, 58], and much more. Many of these problems are nonconvex, but even in the realm of convexity, eigenvalue optimization has a prominent place. Semidefinite programming (SDP) is an important class of convex programs, which may be solved by way of eigenvalue optimization [48].

The idea of solving large semidefinite programs via eigenvalue optimization can be traced back to [16, 25, 41]. It results from the insight that interior-point methods are not the appropriate choice when problems are sizable. Due to its importance in practice, eigenvalue optimization has been intensively studied since the 1980s. Early contributions are Wolfe [59], Cullum, Donath, and Wolfe [16], Polak and Wardi [54], and Fletcher [19]. Starting in the late 1980s, Overton contributed a series of papers ([49, 50], and [51] with Womersley), where in particular Newton-type methods are discussed. Oustry [48] presents a synthesis of first and second order methods suited for convex maximum eigenvalue functions.

Here our interest is in nonconvex eigenvalue programs, which arise frequently in automatic control applications and especially in controller synthesis. In particular, solving bilinear matrix inequalities (BMIs) is a prominent application, which may be addressed via nonconvex eigenvalue optimization. In [45, 46] we have shown how to adapt the approach of [41, 48] to handle nonconvex situations. Applications and extensions of these ideas are presented in [4, 1, 57, 10, 5, 6].

*Received by the editors March 22, 2006; accepted for publication (in revised form) December 4, 2007; published electronically March 19, 2008. The authors acknowledge financial support from Agence Nationale de Recherche (ANR) under contract SSIA_NV_6 *Controvert* and contract NT05-1_43040 *Guidage*, and from *Fondation d'Entreprise EADS* under contract *Solving challenging problems in feedback control*.

http://www.siam.org/journals/siopt/19-1/66519.html

†CERT-ONERA, 2, avenue Edouard Belin, 31055 Toulouse, France (pierre.apkarian@cert.fr).

‡Université Paul Sabatier, Institut de Mathématiques, 118, route de Narbonne, 31062 Toulouse, France (noll@mip.ups-tlse.fr, oprot@mip.ups-tlse.fr).

The goal of the present paper is twofold. In the first part we investigate how to expand on the idea of Helmberg and Rendl's spectral bundle method [25] in order to deal with nonconvex eigenvalue programs. Nonconvexity requires a new approximation technique, complementing the convex mechanism used in [25]. We achieve our goal by a trust region approach or, what is equivalent, by a dual approach using proximity control. This method has antecedents in classical bundling, such as in Lemaréchal [36, 37, 38], Lemaréchal, Nemirovskii, and Nesterov [40], and Kiwiel [31, 32, 33]. Extensions of the convex case to include bound constraints are given in [23].

In the second part we extend our method to address more general classes of functions which are infinite suprema of maximum eigenvalue functions. This includes optimization of the H_∞-norm, an important application in feedback control synthesis. Optimization of the H_∞-norm has been pioneered by Polak and coworkers. See, for instance, [42, 43, 52], and the references given there. Our own approach to optimizing the H_∞-norm is developed in [4, 1, 5].

The structure of the paper is as follows. After some preparations in sections 2–5, the algorithm is presented in section 6. Convergence analysis follows in sections 7 and 8. The semi-infinite case, which includes optimization of the H_∞-norm, is presented in section 9. While the main objective of this work is the convergence analysis of our method, we have added several numerical tests for eigenvalue programs in section 10 to validate the algorithm. Numerical tests for the H_∞-norm and for related problems are presented in [7].

Notation. Our terminology follows [28] and [14]. We let $\|\cdot\|$ denote the Euclidean norm on the space $\mathbb{R}^n$ equipped with the scalar product $x^\top y$, while the space $\mathbb{S}^m$ of $m \times m$ symmetric matrices is equipped with the scalar product $X \bullet Y = \mathrm{Tr}(XY)$. The corresponding matrix norm is also denoted by $\|\cdot\|$. For $X \in \mathbb{S}^m$, $X \preceq 0$ means X is negative semidefinite.

2. Elements from nonsmooth analysis. Recall that the maximum eigenvalue function $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is convex but generally nonsmooth and defined on the space $\mathbb{S}^m$ of $m \times m$ symmetric or Hermitian matrices. We consider composite functions of the form $f = \lambda_1 \circ F$, where $F : \mathbb{R}^n \to \mathbb{S}^m$ is a class C^2 operator. Notice that f is nonsmooth, due to nonsmoothness of λ_1, and nonconvex unless F is an affine operator. The case where F is affine and therefore f convex has been studied by many authors [16, 25, 41]. Here our interest is focused on handling nonconvex f.

Notice that as a composite function, f has a favorable structure. In particular, the Clarke subdifferential [14] is given by the chain rule

$$\partial f(x) = F'(x)^\star \partial\lambda_1\left(F(x)\right),$$

where $\partial\lambda_1(X)$ is the usual subdifferential of convex analysis at $X \in \mathbb{S}^m$, and where $F'(x)$ is the derivative of F, $F'(x)^\star$ its adjoint, mapping $\mathbb{S}^m$ back into $\mathbb{R}^n$. Recall that λ_1 is itself highly structured, as it is the support function of the convex compact set

$$\mathcal{G} = \{G \in \mathbb{S}^m : G \succeq 0, \mathrm{Tr}(G) = 1\}.$$

That means $\lambda_1(X) = \max\{G \bullet X : G \in \mathcal{G}\}$, and therefore (cf. [28, I, sect. 5.1, p. 275])

$$\partial\lambda_1(X) = \{G \in \mathcal{G} : G \bullet X = \lambda_1(X)\}.$$

3. First local model. Consider the minimization of $f = \lambda_1 \circ F$ over $\mathbb{R}^n$ and suppose $x \in \mathbb{R}^n$ is the current iterate. In order to generate a descent step from x to y, we may consider the following convex model of f around x:

$$\phi(y;x) = \lambda_1 \left(F(x) + F'(x)(y-x)\right), \tag{1}$$

where $y \mapsto F(x) + F'(x)(y-x)$ is the first order affine approximation of $F(y)$ in a neighborhood of x. Clearly $\phi(x;x) = f(x)$, and $f = \phi$ in those cases where F itself is affine (see, e.g., [16, 48, 25]). Taylor's theorem suggests that $\phi(y;x) \approx f(y)$ for y sufficiently close to x. This observation is made precise by the following.

PROPOSITION 1. *For every bounded set $B \subset \mathbb{R}^n$ there exists a constant $L > 0$ such that*

$$|f(y) - \phi(y;x)| \leq L\|y-x\|^2 \tag{2}$$

for all $x, y \in B$.

Proof. Notice that for any given matrices $A, E \in \mathbb{S}^m$, the estimate

$$\lambda_m(E) \leq \lambda_1(A+E) - \lambda_1(A) \leq \lambda_1(E)$$

is satisfied. This is also known as Weyl's theorem. Now as F is of class C^2, expanding at $x \in B$ gives $F(y) = F(x) + F'(x)(y-x) + R(y;x)$ with $\|R(y;x)\| \leq L\|y-x\|^2$ for some constant $L > 0$ and all $x, y \in B$. Using $X = F(x)$, $D = F'(x)(y-x)$, we have $f(y) = \lambda_1 \left(X + D + R(y;x)\right)$. We now apply Weyl's theorem with $A = X + D$, $E = R(y;x)$, which gives

$$\begin{aligned} |f(y) - \phi(y;x)| &= |\lambda_1(X+D) - \lambda_1(X+D+R(y;x))| \\ &\leq \max\left\{|\lambda_1\left(R(y;x)\right)|, |\lambda_m\left(R(y;x)|\right\}\right. \\ &\leq \|R(y;x)\| \leq L\|y-x\|^2 \end{aligned}$$

for all $x, y \in B$. That proves the claim. ∎

4. Second local model. Along with (1) we consider a second local model of f in a neighborhood of the current iterate x, which we update recursively. Notice that

$$\phi(y;x) = \max\left\{G \bullet \left[F(x) + F'(x)(y-x)\right] : G \in \mathcal{G}\right\},$$

where $\mathcal{G} = \{G \in \mathbb{S}^m : \mathrm{Tr}(G) = 1, G \succeq 0\}$ as before. This suggests the following approximation $\phi_k(y;x)$ of f, where $\mathcal{G}$ is replaced by a smaller and easier-to-compute subset $\mathcal{G}_k \subset \mathcal{G}$. We will generate a sequence $\mathcal{G}_k \subset \mathcal{G}$ of such approximations and let

$$\phi_k(y;x) = \max\left\{G \bullet \left[F(x) + F'(x)(y-x)\right] : G \in \mathcal{G}_k\right\}. \tag{3}$$

Clearly $\phi_k \leq \phi$. The idea is that we generate descent steps for the $\phi_k(\cdot;x)$, which will ultimately lead to descent in f at x, as the agreement between f and ϕ_k improves at each step k. The minimal requirement for $\mathcal{G}_k$ is the following and is obvious.

LEMMA 2. *Suppose $\mathcal{G}_k$ contains a subgradient of the form $G = ee^\top \in \mathcal{G}$, where e is a normalized eigenvector associated with $\lambda_1\left(F(x)\right)$. Then $\phi_k(x;x) = f(x)$.*

5. Proximity control. Let x be our current iterate. In order to generate a new trial step, we use the current model $\phi_k(\cdot;x)$ and compute the solution y^{k+1} of the unconstrained optimization program,

$$\text{minimize} \quad \phi_k(y;x) + \tfrac{\tau_k}{2}\|y-x\|^2, \quad y \in \mathbb{R}^n, \tag{4}$$

where $\tau_k \geq 0$ is the proximity parameter, and where the term $\frac{\tau_k}{2}\|y-x\|^2$ is referred

to as the proximity control. It is well known (see, e.g., [28, II, Prop. 2.2.3, p. 291]) that (4) is equivalent to a trust region program of the form

$$\begin{aligned} &\text{minimize} && \phi_k(y;x), \quad y \in \mathbb{R}^n \\ &\text{subject to} && \|y-x\| \le t_k, \end{aligned} \tag{5}$$

where $t_k > 0$ is the trust region radius. Indeed, minima of (5) and minima of (4) are in one-to-one correspondence in the following sense: If y^{k+1} is a minimum of (4) for fixed $\tau_k > 0$, then y^{k+1} also solves (5) with $t_k := \|y^{k+1} - x\|$ and associated multiplier $\tau_k > 0$. Conversely, if y^{k+1} solves (5) and if the associated multiplier τ_k is strictly positive, then y^{k+1} solves (4) with that proximity parameter $\tau_k > 0$. The case $\tau_k = 0$ in (4) obviously corresponds to those cases in (5) where the trust region constraint is inactive.

With the models ϕ and ϕ_k we introduce two levels of approximation of f, so it is not surprising that two mechanisms to adjust the degree of exactness are applied. First, in order to control the agreement between f and ϕ, we need to adjust t_k at each step, which is done indirectly via the management of τ_k. If the agreement between f and ϕ is good, we increase t_k, which corresponds to decreasing τ_k, while we have to reduce t_k when agreement is bad, achieved indirectly by increasing τ_k. Second, we update $\mathcal{G}_k$ into $\mathcal{G}_{k+1}$ after each trial y^{k+1} in order to drive ϕ_k closer to ϕ, and thereby also closer to f. We use the standard terminology in nonsmooth optimization. If the solution y^{k+1} of (4) is not used as the next iterate, we call it a null step. If y^{k+1} is accepted and becomes the next iterate x^+, we speak of a serious step.

In order to test the quality of the trial steps y^{k+1}, we use the quotient

$$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \phi_k(y^{k+1};x)}. \tag{6}$$

Fixing constants $0 < \gamma < \Gamma < 1$, we say that f and $\phi_k(\cdot;x)$ are in good agreement when $\rho_k > \Gamma$, and we say that the agreement is bad if $\rho_k < \gamma$. The bad case includes, in particular, situations where $\rho_k \le 0$. Since always $f(x) - \phi_k(y^{k+1};x) > 0$, unless $0 \in \partial f(x)$, we deduce that $\rho_k \le 0$ corresponds to cases where $f(x) - f(y^{k+1}) \le 0$, that is, where the proposed step y^{k+1} is not even a descent step for f. In our algorithm we use the following rule: y^{k+1} is accepted as soon as $\rho_k \ge \gamma$, i.e., as soon as the step is not bad. The question is then what we shall do when agreement between f and ϕ_k is bad, i.e., when $\rho_k < \gamma$.

Here we compute a second test parameter,

$$\widetilde{\rho}_k := \frac{f(x) - \phi(y^{k+1};x)}{f(x) - \phi_k(y^{k+1};x)},$$

and we compare it to a second control parameter $\widetilde{\gamma}$, where $\gamma < \widetilde{\gamma} < \frac{1}{2}$. We then have two possibilities. If $\rho_k < \gamma$ and also $\widetilde{\rho}_k < \widetilde{\gamma}$, then we do not change τ_k, but improve the approximation $\mathcal{G}_{k+1}$ so that ϕ_{k+1} gets closer to ϕ. On the other hand, if $\rho_k < \gamma$, but $\widetilde{\rho}_k \ge \widetilde{\gamma}$, then ϕ and f are not in good agreement, while ϕ_k is already close to ϕ. Driving ϕ_k even closer to ϕ in that case alone will therefore not improve the situation. Here we have to decrease the trust region radius t_k, or what comes down to the same, increase the proximity control parameter τ_k. While doing this, we still update $\mathcal{G}_k$ to a better $\mathcal{G}_{k+1}$, i.e., we still let ϕ_k approach ϕ, so this process is always applied.

6. Aggregate subgradients. As our convergence analysis will show, the approximations $\mathcal{G}_k \subset \mathcal{G}$ need only satisfy the following three conditions:

- (G_1) $G_0 = e_0 e_0^\top \in \mathcal{G}_k$ for some normalized eigenvector e_0 associated with $\lambda_1\,(F(x))$.
- (G_2) $G_{k+1} = e_{k+1}e_{k+1}^\top \in \mathcal{G}_{k+1}$ for some normalized eigenvector e_{k+1} associated with $\lambda_1\big(F(x) + F'(x)(y^{k+1} - x)\big)$.
- (G_3) $G_k^* \in \mathcal{G}_{k+1}$ for some of the $G_k^* \in \mathcal{G}_k$, where the maximum $\phi_k(y^{k+1}; x)$ is attained, and which satisfies $0 = F'(x)^\star G_k^* + \tau_k(y^{k+1} - x)$.

Below we will discuss practical choices of the sets $\mathcal{G}_k$, combining ideas from [25], [48], [24], and [45]. We let $\mathcal{G}_k$ consist of sets of the form

$$\alpha_k \overline{G}_k + Q_k Y_k Q_k^\top, \tag{7}$$

where $Y_k \in \mathbb{S}^{r_k}$ has $Y_k \succeq 0$, $\alpha_k + \mathrm{Tr}(Y_k) = 1$, $0 \le \alpha_k \le 1$, where Q_k is an $m \times r_k$ matrix whose $r_k \ge 1$ columns form an orthogonal basis of an invariant subspace of $F(x) + F'(x)(y^{k+1} - x)$, and where $\overline{G}_k \in \mathcal{G}$ is the aggregate subgradient. We assume that at least one normalized eigenvector e_k associated with the maximum eigenvalue $\lambda_1\big(F(x) + F'(x)(y^k - x)\big)$ is in the span of the columns of Q_k, and moreover, that e_0 is in the span of the columns of Q_k at all times. The idea is to build the new set $\mathcal{G}_{k+1}$ along the same lines, using an updating strategy, which we now explain.

Let y^{k+1} be the solution of program (4), obtained with the help of $\mathcal{G}_k$, and suppose it is a null step. The necessary optimality condition gives $0 \in \partial\phi_k(y^{k+1}; x) + \tau_k(y^{k+1} - x)$. Due to the structure of ϕ_k and (7), this means there exist $G_k^* \in \mathcal{G}_k$ such that

$$0 = F'(x)^\star G_k^* + \tau_k(y^{k+1} - x), \quad G_k^* = \alpha_k^* \overline{G}_k + Q_k Y_k^* Q_k^\top, \tag{8}$$

where $0 \le \alpha_k^* \le 1$, $Y_k^* \succeq 0$, and $\alpha_k^* + \mathrm{Tr}(Y_k^*) = 1$. Now the simplest method is to let

$$\overline{G}_{k+1} = \alpha_k^* \overline{G}_k + Q_k Y_k^* Q_k^\top, \tag{9}$$

the new aggregate subgradient. Helmberg and Rendl [25] use a refinement of (9), which is suited for large problem size: Let $Y_k^* = PDP^\top$ be a spectral decomposition of the $r_k \times r_k$ matrix Y_k^*. Decompose $P = [P_1 P_2]$ with corresponding spectra D_1 and D_2 so that P_1 contains as columns those eigenvectors associated with the large eigenvalues of Y_k^*, and P_2 are the remaining columns. Now put

$$\overline{G}_{k+1} = \big(\alpha_k^* \overline{G}_k + Q_k P_2 D_2 P_2^\top Q_k^\top\big) \,/\, (\alpha_k^* + \mathrm{Tr}(D_2))\,, \tag{10}$$

the new aggregate subgradient, which is an element of $\mathcal{G}$. In this way only the minor part of Y_k^* is kept in the aggregate subgradient. The dominant part of Y_k^* is retained in the next eigenbasis by letting $Q_k P_1$ be part of Q_{k+1}. Moreover, in view of axiom (G_2), one eigenvector e_{k+1} of the maximum eigenvalue of $F(x) + F'(x)(y^{k+1} - x)$ is computed and included in Q_{k+1}. In order to guarantee axiom (G_1), we also keep at least one normalized eigenvector e_0 associated with the maximum eigenvalue of $F(x)$ in Q_{k+1}.

Altogether, $\mathcal{G}_{k+1}$ consists of all $\alpha\overline{G}_{k+1} + Q_{k+1}Y_{k+1}Q_{k+1}^\top$, where $0 \le \alpha \le 1$, $Y_{k+1} \succeq 0$, and $\alpha + \mathrm{Tr}(Y_{k+1}) = 1$, and where Q_{k+1} has the properties above. For this construction we have the following.

LEMMA 3. *The sets $\mathcal{G}_k$ so defined satisfy the rules (G_1)–(G_3). In particular, $\phi_k(x; x) = f(x)$, $\phi_{k+1}(y^{k+1}; x) = \phi(y^{k+1}; x)$, $\phi_{k+1}(y^{k+1}; x) \ge \phi_k(y^{k+1}; x)$, and condition* (8) *hold for every k.*

Remark. In a traditional bundle method we would refer to $\overline{g}_k = F'(x)^\star \overline{G}_k$ as the aggregate subgradient. Here we use the term aggregate for both $\overline{g}_k$ and $\overline{G}_k$ because

there is no risk of ambiguity. But are both elements really needed? The authors of [25] point out that storing $\bar{g}_k$ is cheaper than storing $\overline{G}_k$, so naturally they work in g-space and not in G-space.

Now observe an important difference of our present case with the convex case in [25], where $F'(x)^* = A^*$ is independent of x. Since our $F'(x)^*$ depends on x, as soon as a serious step $x \to x^+$ is taken, $\bar{g}_k = F'(x)^*\overline{G}_k$ is no longer useful at x^+, because it is no longer a subgradient of f at x^+. However, $\overline{G}_k$ is still useful. It suffices to replace $\bar{g}_k$ by $\bar{g}_k^+ := F'(x^+)^*\overline{G}_k$, which is a subgradient for f at x^+. So if we want to "recycle" old aggregates in the next serious loop, we have to store $\overline{G}_k$ and not $\bar{g}_k$. On the other hand, if we are happy to keep aggregates only within one single inner loop, then we do not need $\overline{G}_k$ and our case is similar to [25].

Remark. Conditions (G_1)–(G_3) leave a lot of freedom for the choice of the bases Q_k. In [24] Helmberg and Oustry investigate the convex case and discuss ways to combine their two approaches [25] and [48] into a unified method. An alternative approach is Polak and Wardi [54]. For nonconvex eigenvalue functions we have proposed in [45, 46] an extension of Oustry's approach.

Spectral bundle algorithm for $\min_{x\in\mathbb{R}^n} f(x)$.

Parameters: $0 < \gamma < \widetilde{\gamma} < \Gamma < 1$.

0. **Initialize outer loop.** Find initial iterate x and compute $f(x)$.
1. **Outer loop.** Stop if $0 \in \partial f(x)$ at current outer iterate x. Otherwise goto inner loop.
2. **Initialize inner loop.** Let $\mathcal{G}_1 \subset \partial\lambda_1(F(x))$, $\overline{G}_1 = \frac{1}{m} I_m$, put inner loop counter $k = 1$, and choose $\tau_1 > 0$. If old value for τ from previous sweep is memorized, use it to initialize τ_1.
3. **Tangent program.** At counter k with given $\tau_k > 0$ and $\mathcal{G}_k$ solve
$$\min_{y\in\mathbb{R}^n} \phi_k(y;x) + \frac{\tau_k}{2}\|y - x\|^2.$$
Solution is y^{k+1}. Find $G_k^* \in \mathcal{G}_k$ where (3) at y^{k+1} is attained. Write $G_k^* = \alpha_k^*\overline{G}_k + Q_k Y_k Q_k^\top$ according to (7).
4. **Acceptance test.** Compute $f(y_{k+1})$ and check whether
$$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \phi_k(y^{k+1};x)} \geq \gamma.$$
If this is the case, put $x^+ = y^{k+1}$ (serious step). Compute new memory element τ^+ as
$$\tau^+ = \begin{cases} \dfrac{\tau_k}{2} & \text{if } \rho_k > \Gamma, \\ \tau_k & \text{else} \end{cases}$$
Then go back to step 1 to commence a new sweep of outer loop. On the other hand, if $\rho_k < \gamma$, then continue inner loop with step 5.
5. **Agreement test.** Compute $\phi(y^{k+1};x)$ and control parameter
$$\widetilde{\rho}_k = \frac{f(x) - \phi(y^{k+1};x)}{f(x) - \phi_k(y^{k+1};x)}.$$
Put
$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \rho_k < \gamma \text{ and } \widetilde{\rho}_k < \widetilde{\gamma}, \\ 2\tau_k & \text{if } \rho_k < \gamma \text{ and } \widetilde{\rho}_k \geq \widetilde{\gamma}. \end{cases}$$
6. **Aggregate subgradient.** Compute new set $\mathcal{G}_{k+1}$ according to (7), with (G_1)–(G_3) satisfied. New aggregate subgradient is $\overline{G}_{k+1} = G_k^*$.
7. **Inner loop.** Increase counter $k \to k+1$ and go back to step 3.

7. Convergence analysis of inner loop. We have to show that the inner loop is finite, that is, finds a trial point y^{k+1} accepted in step 4 after a finite number k of steps. We prove this by showing that if the inner loop turns forever, that is, $\rho_k < \gamma$ for all k, then $0 \in \partial f(x)$. (Since the inner loop is not entered when $0 \in \partial f(x)$, this is an argument by contradiction.) There are two subcases to be discussed, depending on the decision in step 5. These will be addressed in Lemmas 4 and 5.

Our first concern is when $\rho_k < \gamma$ but $\widetilde{\rho}_k \geq \widetilde{\gamma}$. This is indeed the situation where we are far from the convex case. Namely, $\widetilde{\rho}_k \geq \widetilde{\gamma}$ means that ϕ_k is in good agreement with ϕ, but unfortunately $\rho_k < \gamma$ says that ϕ is not a good model of f, which is usually due to the fact that f is nonconvex in a neighborhood of the current x. In consequence, ϕ_k cannot be expected to be a good model of f either. This is addressed in step 5 of the algorithm by increasing the proximity parameter τ_k, which as we know is equivalent to reducing the trust region radius. This is the only way to improve the agreement between ϕ and f.

LEMMA 4. *Suppose the algorithm generates an infinite sequence of trial steps y^{k+1} such that always $\rho_k < \gamma$. Then $\widetilde{\rho}_k < \widetilde{\gamma}$ for some k_0 and all $k \geq k_0$.*

Proof. (i) Assume on the contrary that $\rho_k < \gamma$ for all k, but at the same time $\widetilde{\rho}_k \geq \widetilde{\gamma}$ for infinitely many $k \in \mathbb{N}$. Then according to the update rule in step 5 of the algorithm, the sequence τ_k tends to $+\infty$. As a consequence of the necessary optimality condition we have $0 \in \partial\phi_k(y^{k+1}; x) + \tau_k(y^{k+1} - x)$. Now observe that due to the special form (7), the subgradients of all functions $\phi_k(\cdot; x)$ are uniformly bounded by $\|F'(x)^\star\|$. Given that $\tau_k \to \infty$, we then must have $y^{k+1} \to x$. Using $\tau_k(x - y^{k+1}) \in \partial\phi_k(y^{k+1}; x)$ calls for the subgradient inequality, which gives

$$\tau_k \left(x - y^{k+1}\right)^\top \left(x - y^{k+1}\right) \leq \phi_k(x; x) - \phi_k(y^{k+1}; x) = f(x) - \phi_k(y^{k+1}; x),$$

the latter by Lemma 3. In other words,

$$\frac{\tau_k \|x - y^{k+1}\|^2}{f(x) - \phi_k(y^{k+1}; x)} \leq 1. \tag{11}$$

(ii) Now we expand the test parameters as follows:

$$\begin{aligned}
\widetilde{\rho}_k &= \rho_k + \frac{f(y^{k+1}) - \phi(y^{k+1}; x)}{f(x) - \phi_k(y^{k+1}; x)} \\
&\leq \rho_k + \frac{L\|x - y^{k+1}\|^2}{f(x) - \phi_k(y^{k+1}; x)} \qquad \text{(using Proposition 1)} \\
&\leq \rho_k + \frac{L}{\tau_k} \qquad \text{(using (11))}.
\end{aligned}$$

Since $\tau_k \to \infty$, we deduce $\limsup_{k\to\infty} \widetilde{\rho}_k \leq \limsup_{k\to\infty} \rho_k \leq \gamma$, contradicting $\widetilde{\rho}_k \geq \widetilde{\gamma} > \gamma$ for infinitely many k. ∎

Remark. Notice that the proof of Lemma 4 uses Lemma 3, which in turn exploits axiom (G_1). Axioms (G_2) and (G_3) will be needed in the next lemma.

As a consequence of Lemma 4 we see that the algorithm, when faced with the bad case $\rho_k < \gamma$, will continue to increase τ_k, until eventually $\widetilde{\rho}_k < \widetilde{\gamma}$, too. From some index k_0 onwards, we will then be in the first case in step 5 of the algorithm, where the parameter τ_k is frozen, i.e., $\tau_k =: \tau$ for $k \geq k_0$. We then have no easy argument to deduce $y^{k+1} \to x$. Here, indeed, we will have to exploit properties (G_2) and (G_3) of the update rule $\mathcal{G}_k \to \mathcal{G}_{k+1}$. We follow the line of [25, Lemma 4.2].

LEMMA 5. *Let x be the current iterate and suppose the algorithm generates an infinite sequence of trial steps y^{k+1}, where $\rho_k < \gamma$ for all k while $\widetilde{\rho}_k < \widetilde{\gamma}$ for some k_0 and all $k \geq k_0$. Then $0 \in \partial f(x)$.*

Proof. (i) As we mentioned already, $\rho_k < \gamma$ and $\widetilde{\rho}_k < \gamma$ for all $k \geq k_0$ implies that in step 5 of the algorithm, $\tau_k =: \tau$ is frozen for $k \geq k_0$. A priori we therefore do not know whether $y^{k+1} \to x$, as we did in the proof of Lemma 4. This complicates the following analysis.

(ii) Let us introduce the function

$$\psi_k(y;x) = \phi_k(y;x) + \tfrac{\tau}{2}\|y-x\|^2;$$

then by its definition, y^{k+1} is the global minimum of $\psi_k(\cdot;x)$ for $k \geq k_0$. Let $G_k^* \in \mathcal{G}_k$ be the subgradient where the supremum $\phi_k(y^{k+1};x)$ is attained and which is retained in $\mathcal{G}_{k+1}$ in accordance with rule (G_3) and also with step 3 of the algorithm. That means

$$\phi_k(y^{k+1};x) = G_k^* \bullet \left[F(x) + F'(x)(y^{k+1}-x)\right], \tag{12}$$

and also

$$\psi_k(y^{k+1};x) = G_k^* \bullet \left[F(x) + F'(x)(y^{k+1}-x)\right] + \tfrac{\tau}{2}\|y^{k+1}-x\|^2. \tag{13}$$

We introduce the function

$$\psi_k^*(y;x) = G_k^* \bullet [F(x) + F'(x)(y-x)] + \tfrac{\tau}{2}\|y-x\|^2.$$

Then $\psi_k^*(y^{k+1};x) = \psi_k(y^{k+1};x)$ and

$$\psi_k^*(y;x) \leq \psi_{k+1}(y;x) \tag{14}$$

for $k \geq k_0$, because $G_k^* \in \mathcal{G}_{k+1}$. We claim that

$$\psi_k^*(y;x) = \psi_k^*(y^{k+1};x) + \tfrac{\tau}{2}\|y-y^{k+1}\|^2. \tag{15}$$

An easy way to see this is to observe that ψ_k^* is quadratic and expand it, using $\nabla\psi_k^*(y;x) = F'(x)^\star G_k^* + \tau(y-x)$ and $\nabla^2\psi_k^*(y;x) = \tau I$. Then clearly,

$$\psi_k^*(y;x) = \psi_k^*(y^{k+1};x) + \nabla\psi_k^*(y^{k+1};x)^\top(y-y^{k+1}) + \tfrac{\tau}{2}(y-y^{k+1})^\top(y-y^{k+1}).$$

Formula (15) will therefore be established as soon as we show that the first order term in this expansion vanishes. But this term is

$$\begin{aligned}
&\nabla\psi_k^*(y^{k+1};x)^\top(y-y^{k+1}) \\
&\quad = \left(F'(x)^\star G_k^*\right)^\top(y-y^{k+1}) + \tau(y^{k+1}-x)^\top(y-y^{k+1}) \\
&\quad = \tau(x-y^{k+1})^\top(y-y^{k+1}) + \tau(y^{k+1}-x)^\top(y-y^{k+1}) \quad \text{(using (8))} \\
&\quad = 0.
\end{aligned}$$

That proves formula (15).

(iii) From (ii) we have

$$\begin{aligned}
\psi_k(y^{k+1};x) &\le \psi_k^*(y^{k+1};x) + \tfrac{\tau}{2}\|y^{k+2}-y^{k+1}\|^2 \quad (\text{using } \psi_k^*(y^{k+1};x) = \psi_k(y^{k+1};x)) \\
&= \psi_k^*(y^{k+2};x) \qquad (\text{using (15)}) \\
&\le \psi_{k+1}(y^{k+2};x) \qquad (\text{using (14)}) \\
&\le \psi_{k+1}(x;x) \qquad (y^{k+2} \text{ is minimizer of } \psi_{k+1}) \qquad (16) \\
&= \phi_{k+1}(x;x) \le \phi(x;x).
\end{aligned}$$

We deduce that the sequence $\psi_k(y^{k+1};x)$ is monotonically increasing and bounded above by $\phi(x;x)$. It therefore converges to some $\psi^* \le \phi(x;x)$.

Going back to (16) with this information, we see that the term $\frac{\tau}{2}\|y^{k+2}-y^{k+1}\|^2$ is now squeezed in between two convergent terms with the same limit ψ^*, and must therefore tend to zero. Consequently, $\|y^{k+1}-x\|^2-\|y^{k+2}-x\|^2$ also tends to 0, because the sequence y^k is bounded. (Boundedness of the y^{k+1} was already used in the proof of the previous lemma and follows from the particular form (7) of the subgradients and the fact that the sequence τ_k is nondecreasing and therefore bounded away from 0.)

Recalling $\phi_k(y;x) = \psi_k(y;x) - \frac{\tau}{2}\|y-x\|^2$, we deduce, using both convergence results, that

$$\begin{aligned}
&\phi_{k+1}(y^{k+2};x) - \phi_k(y^{k+1};x) \\
&\qquad = \psi_{k+1}(y^{k+2};x) - \psi_k(y^{k+1};x) - \tfrac{\tau}{2}\|y^{k+2}-x\|^2 + \tfrac{\tau}{2}\|y^{k+1}-x\|^2 \to 0. \qquad (17)
\end{aligned}$$

(iv) Let e_{k+1} be the normalized eigenvectors of $F(x)+F'(x)(y^{k+1}-x)$ associated with λ_1, which we pick in step 5 of the algorithm and according to rule (G_2). Then $g_k = F'(x)^\star e_{k+1}e_{k+1}^\top$ is a subgradient of $\phi_{k+1}(\cdot;x)$ at y^{k+1}. Hence by the subgradient inequality

$$\phi_{k+1}(y^{k+1};x) + g_k^\top\left(y-y^{k+1}\right) \le \phi_{k+1}(y;x).$$

Since $\phi_{k+1}(y^{k+1};x) = \phi(y^{k+1};x)$ by Lemma 3, respectively, rule (G_2), we have the estimate

$$\phi(y^{k+1};x) + g_k^\top\left(y-y^{k+1}\right) \le \phi_{k+1}(y;x). \qquad (18)$$

Now observe that

$$\begin{aligned}
0 &\le \phi(y^{k+1};x) - \phi_k(y^{k+1};x) \\
&= \phi(y^{k+1};x) + g_k^\top\left(y^{k+2}-y^{k+1}\right) - \phi_k(y^{k+1};x) - g_k^\top\left(y^{k+2}-y^{k+1}\right) \\
&\le \phi_{k+1}(y^{k+2};x) - \phi_k(y^{k+1};x) + \|g_k\|\left\|y^{k+2}-y^{k+1}\right\| \qquad (\text{using (18)}),
\end{aligned}$$

and this term tends to 0 due to (17), because $y^{k+2}-y^{k+1} \to 0$, and because the sequence g_k is bounded. We deduce that $\phi(y^{k+1};x) - \phi_k(y^{k+1};x) \to 0$.

(v) We now show that $\phi_k(y^{k+1};x) \to f(x)$, and then of course also $\phi(y^{k+1};x) \to f(x)$. Assume, contrary to what is claimed, that $\limsup_{k\to\infty} f(x) - \phi_k(y^{k+1};x) =: \eta > 0$. Choose $\delta > 0$ such that $\delta < (1-\widetilde{\gamma})\eta$. It follows from part (iv) that there exists $k_1 \geq k_0$ such that

$$\phi(y^{k+1};x) - \delta \leq \phi_k(y^{k+1};x) \tag{19}$$

for all $k \geq k_1$. Using $\widetilde{\rho}_k \leq \widetilde{\gamma}$ for $k \geq k_1$ then gives

$$\begin{aligned} \widetilde{\gamma}\left(\phi_k(y^{k+1};x) - f(x)\right) &\leq \phi(y^{k+1};x) - f(x) \\ &\leq \phi_k(y^{k+1};x) + \delta - f(x), \end{aligned} \tag{20}$$

which implies $\widetilde{\gamma}\eta \geq \eta - \delta$. This contradicts the choice of δ and therefore shows $\eta = 0$.

(vi) Having shown $\phi(y^{k+1};x) \to f(x)$ and $\phi_k(y^{k+1};x) \to f(x)$, we argue that $y^{k+1} \to x$. This follows from the definition of y^{k+1}, because

$$\psi_k(y^{k+1};x) = \phi_k(y^{k+1};x) + \tfrac{\tau}{2}\|y^{k+1}-x\|^2 \leq \psi_k(x;x) = \phi_k(x;x) = f(x).$$

Since $\phi_k(y^{k+1};x) \to f(x)$, we have indeed $\|y^{k+1}-x\| \to 0$ by a sandwich argument.

To finish the proof, let us show that $0 \in \partial f(x)$. Notice first that the necessary optimality condition gives $0 \in \partial\psi_k(y^{k+1};x) = \partial\phi_k(y^{k+1};x) + \tau(y^{k+1}-x)$, which implies

$$\tau(x - y^{k+1}) \in \partial\phi_k(y^{k+1};x).$$

The subgradient inequality gives

$$\begin{aligned} \tau(x-y^{k+1})^\top(y-y^{k+1}) &\leq \phi_k(y;x) - \phi_k(y^{k+1};x) \\ &\leq \phi(y;x) - \phi_k(y^{k+1};x) \quad (\text{using } \phi_k \leq \phi) \end{aligned}$$

for every y. Passing to the limit, observing $\tau(y^{k+1}-x) \to 0$ and $\phi_k(y^{k+1};x) \to \phi(x;x)$, we obtain the estimate

$$0 \leq \phi(y;x) - \phi(x;x)$$

for every y, which by convexity of $\phi(\cdot;x)$ implies $0 \in \partial\phi(x;x)$. Since $\partial\phi(x;x) = \partial f(x)$, we have shown $0 \in \partial f(x)$, as claimed. ∎

Remark. Various modifications of our algorithm may be considered. For instance, whenever a null step y^{k+1} is made, that is, $\rho_k < \gamma$, we should first check whether y^{k+1} gives descent in f:

$$f(x) - f(y^{k+1}) \geq \delta_1 > 0. \tag{21}$$

If this is not the case, the trust region radius is certainly too large, so we should increase τ_k right away. As presented, this will also happen, but after several null steps, bringing ϕ_k closer to ϕ, until the criterion in step 4 is met.

In the same vein, even when y^{k+1} gives descent in f, but slightly, so that $\rho_k > \gamma$ fails, we may check whether

$$\sigma_k := \frac{f(x) - f(y^{k+1})}{f(x) - \phi(y^{k+1};x)} \geq \delta_2 \tag{22}$$

for some $\frac{1}{2} < \delta_2 < 1$. If $\sigma_k < \delta_2$, then f and ϕ are not in good agreement. In this case, our algorithm will keep τ_k fixed and will start driving ϕ_k closer to ϕ. Eventually this will lead to a moment where $\widetilde{\rho}_k \geq \widetilde{\gamma}$, and then τ_k will be increased to $2\tau_k$. As above one may argue that in this case we have lost time, because we have brought ϕ_k closer to ϕ even though ϕ is too far away from f, only to notice in the end that we could not avoid increasing τ_k. This loss of time and energy could be avoided by adding the test (22). If $\sigma_k < \delta_2$, then we increase τ_k right away but keep incrementing ϕ_k. The convergence analysis of this is covered by the two central lemmas of this section.

8. Convergence analysis of the outer loop. All we have to do now is piece things together and show subsequence convergence of the sequence of serious steps x^j retained in the outer loop. We have the following.

THEOREM 6. *Let $f = \lambda_1 \circ F$ be a maximum eigenvalue function and let $x^1 \in \mathbb{R}^n$ be such that the set $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is compact. Then every accumulation point of the sequence x^j of serious steps generated by the algorithm is a critical point of f.*

Proof. (i) From the previous section we know that the inner loop always ends after a finite number of steps k with a new x^+ satisfying the acceptance test, unless we have finite termination due to $0 \in \partial f(x)$. Excluding this case, let us assume that x^j is the sequence of serious steps, satisfying the acceptance test in step 4 of the algorithm. Since y^{k+1} accepted in step 4 becomes the new x^{j+1}, that means

$$f(x^j) - f(x^{j+1}) \geq \gamma \left(f(x^j) - \phi_{k_j}(x^{j+1}; x^j) \right), \tag{23}$$

where j is the counter of the outer loop, k the counter of the inner loop, and where at the outer step j the inner loop was stopped at $k = k_j$. Now recall from the construction that $\tau_{k_j}\left(x^j - x^{j+1}\right) \in \partial \phi_{k_j}(x^{j+1}; x^j)$. The subgradient inequality for $\phi_{k_j}(\cdot; x^j)$ at x^{j+1} therefore gives

$$\tau_{k_j}\left(x^j - x^{j+1}\right)^\top (x^j - x^{j+1}) \leq \phi_{k_j}(x^j; x^j) - \phi_{k_j}(x^{j+1}; x^j) = f(x^j) - \phi_{k_j}(x^{j+1}; x^j),$$

using $\phi_{k_j}(x^j; x^j) = f(x^j)$. That means

$$\tau_{k_j}\|x^{j+1} - x^j\|^2 \leq f(x^j) - \phi_{k_j}(x^{j+1}; x^j) \leq \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right)$$

using (23). Summing up from $j = 1$ to $j = J - 1$ gives

$$\sum_{j=1}^{J-1} \tau_{k_j}\|x^{j+1} - x^j\|^2 \leq \gamma^{-1} \sum_{j=1}^{J-1} f(x^j) - f(x^{j+1}) = \gamma^{-1}\left(f(x^1) - f(x^J)\right),$$

which is bounded above due to the hypothesis that $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is bounded. We deduce convergence of the series

$$\sum_{j=1}^{\infty} \tau_{k_j}\|x^{j+1} - x^j\|^2 < \infty.$$

In particular, $\tau_{k_j}\|x^{j+1} - x^j\|^2 \to 0$.

(ii) Let us prove that this implies $g_j := \tau_{k_j}\left(x^j - x^{j+1}\right) \to 0$, $(j \to \infty)$. Assume on the contrary that there exists an infinite subset $\mathcal{N}$ of $\mathbb{N}$ and some $\mu > 0$ such that $\|g_j\| = \tau_{k_j}\|x^j - x^{j+1}\| \geq \mu > 0$ for every $j \in \mathcal{N}$. In tandem with the summability of $\tau_{k_j}\|x^j - x^{j+1}\|^2$ shown in part (i) this could only mean $x^j - x^{j+1} \to 0$, and at the

same time, $\tau_{k_j} \to \infty$, $j \in \mathcal{N}$. We now argue that there exists yet another infinite subsequence $\mathcal{N}'$ of $\mathbb{N}$ with $\tau_j \to \infty$, $(j \in \mathcal{N}')$, such that for each $j \in \mathcal{N}'$, the doubling rule in step 5 of the algorithm was applied at least once before the step $x^{j+1} = y^{k_j+1}$ was accepted. Indeed, to construct $\mathcal{N}'$ we let, for every $j \in \mathcal{N}$, $j' \le j$ be that outer-loop instant where the τ-parameter was increased for the last time before j, and we let $\mathcal{N}'$ consist of all these j', $j \in \mathcal{N}$. It is possible that $j' = j$, but in general we may have $j' < j$, and we know only that

$$2\tau_{j'-1} \le \tau_{j'} \quad \text{and} \quad \tau_{j'} \ge \tau_{j'+1} \ge \cdots \ge \tau_j.$$

However, since $\tau_j \to \infty$, $j \in \mathcal{N}$, we know that we must have $\tau_{j'} \to \infty$, $j' \in \mathcal{N}'$. Since the doubling rule was applied at least once at the outer-loop counter j', $\mathcal{N}'$ is as claimed.

Let us say that for $j \in \mathcal{N}'$ the doubling rule was applied for the last time at stage $\tau_{k_j-\nu_j}$ for some $\nu_j \ge 1$. That means, $\tau_{k_j-\nu_j+1} = 2\tau_{k_j-\nu_j}$, while the τ-parameter remained unchanged during the following inner steps before acceptance:

$$\tau_{k_j} = \tau_{k_j-1} = \cdots = \tau_{k_j-\nu_j+1} = 2\tau_{k_j-\nu_j}. \tag{24}$$

Now recall that in step 5 of the algorithm we have $\rho_k < \gamma$ and $\widetilde{\rho}_k \ge \widetilde{\gamma}$ for those k where the trial step was not accepted and the doubling rule was applied. Since this is the case at stage $k_j - \nu_j$ we have

$$\rho_{k_j-\nu_j} = \frac{f(x^j) - f(y^{k_j-\nu_j+1})}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} < \gamma$$

and

$$\widetilde{\rho}_{k_j-\nu_j} = \frac{f(x^j) - \phi(y^{k_j-\nu_j+1}; x^j)}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} \ge \widetilde{\gamma}.$$

By (24) we now have

$$\frac{1}{2}\tau_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right) \in \partial\phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j).$$

Using the subgradient inequality for $\phi_{k_j-\nu_j}(\cdot; x^j)$ at $y^{k_j-\nu_j+1}$ and $\phi_{k_j-\nu_j}(x^j; x^j) = f(x^j)$, we obtain

$$\begin{aligned}\frac{1}{2}\tau_{k_j}\left(x^j - y^{k_j-\nu_j+1}\right)^\top\left(x^j - y^{k_j-\nu_j+1}\right) &\le \phi_{k_j-\nu_j}(x^j; x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)\\ &= f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j),\end{aligned}$$

which could also be written as

$$\frac{\tau_{k_j}\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} \le 2. \tag{25}$$

Substituting (25) into the expression for $\widetilde{\rho}_{k_j-\nu_j}$ and expanding gives

$$\begin{aligned}\widetilde{\rho}_{k_j-\nu_j} &= \rho_{k_j-\nu_j} + \frac{f(y^{k_j-\nu_j+1}) - \phi(y^{k_j-\nu_j+1}; x^j)}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)}\\ &\le \rho_{k_j-\nu_j} + \frac{L\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}; x^j)} \qquad \text{(using Proposition 1)}\\ &\le \rho_{k_j-\nu_j} + \frac{2L}{\tau_{k_j}} \qquad \text{(using (25))}.\end{aligned}$$

Here Proposition 1 is applied to the set B of all x^j and $y^{k_j-\nu_j+1}$, $j \in \mathcal{N}'$, which is bounded because $\|y^{k_j-\nu_j+1}\| \le \|F'(x^j)^\star\|$ due to (7), and because the serious steps x^j belong to the level set $\{x \in \mathbb{R}^n : f(x) \le f(x^1)\}$, which is bounded by hypothesis. Since $\rho_{k_j-\nu_j} < \gamma$ and $L/2\tau_{k_j} \to 0$, we have $\limsup_{j\to\infty} \widetilde{\rho}_{k_j-\nu_j} \le \gamma$ in the estimate above, contradicting $\widetilde{\rho}_{k_j-\nu_j} \ge \widetilde{\gamma} > \gamma$ for all $j \in \mathcal{N}'$.

(iii) Having shown that $g_j := \tau_{k_j}(x^j - x^{j+1}) \to 0$, $(j \to \infty)$, let us argue that every accumulation point $\bar{x}$ of the sequence x^j of serious steps must be a critical point. Notice again that since $\{x \in \mathbb{R}^n : f(x) \le f(x^1)\}$ is compact by hypothesis, and since our algorithm is of descent type in the serious steps, the sequence x^j is bounded. Select a convergent subsequence $x^j \to \bar{x}$, $j \in \mathcal{N}$. The same argument applies to the sequence x^{j+1}, $j \in \mathcal{N}$. We may therefore assume that this sequence also has a limit, $\tilde{x}$. Notice that in general we might have $\tilde{x} \ne \bar{x}$. Only in those cases where the τ_{k_j}, $j \in \mathcal{N}$, are bounded away from 0 can we conclude that $x^{j+1} - x^j \to 0$, $j \in \mathcal{N}$. In general, however, according to step 4 of the algorithm, the τ-parameter may very well shrink to 0, and here $x^j - x^{j+1} \to 0$ cannot be assured.

Since g_j is a subgradient of $\phi_{k_j}(\cdot; x^j)$ at $x^{j+1} = y^{k_j+1}$, we have

$$\begin{aligned} g_j^\top h &\le \phi_{k_j}(x^{j+1}+h; x^j) - \phi_{k_j}(x^{j+1}; x^j) \\ &\le \phi(x^{j+1}+h; x^j) - \phi_{k_j}(x^{j+1}; x^j) \qquad (\text{using } \phi_{k_j} \le \phi) \end{aligned}$$

for every test vector h. Now we use the fact that $y^{k_j+1} = x^{j+1}$ was accepted in step 4 of the algorithm, which means

$$\gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) \ge f(x^j) - \phi_{k_j}(x^{j+1}; x^j).$$

Combining these two estimates gives

$$\begin{aligned} g_j^\top h &\le \phi(x^{j+1}+h; x^j) - f(x^j) + f(x^j) - \phi_{k_j}(x^{j+1}; x^j) \\ &\le \phi(x^{j+1}+h; x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right). \end{aligned}$$

Passing to the limit $j \in \mathcal{N}$ and using, in the order named, $g_j \to 0$, $x^{j+1} \to \tilde{x}$, $x^j \to \bar{x}$, and $f(\bar{x}) = \phi(\bar{x}; \bar{x})$, implies

$$0 \le \phi(\tilde{x}+h; \bar{x}) - \phi(\bar{x}; \bar{x})$$

for every test vector h, where the last term $f(x^j) - f(x^{j+1}) \to 0$ by monotonicity. Choosing $h = \bar{x} - \tilde{x} + h'$ therefore implies

$$0 \le \phi(\bar{x}+h'; \bar{x}) - \phi(\bar{x}; \bar{x})$$

for every test vector $h' \in \mathbb{R}^n$, which proves $0 \in \partial\phi(\bar{x}; \bar{x})$. Hence also $0 \in \partial f(\bar{x})$. ∎

In practical tests we observe convergence, and the theoretical possibility of a sequence of iterates with several accumulation points never occurs. This is explained to some extent by the following.

COROLLARY 7. *Suppose $f = \lambda_1 \circ F$ is convex in a closed and bounded neighborhood Ω of x^* and that the iterates (serious steps) x^j remain in Ω. Then the sequence x^j converges to some local minimum $x^\sharp \in \Omega$ with $f(x^*) = f(x^\sharp)$.*

Proof. As the sequence of serious steps x^j satisfies the acceptance condition in step 4 of the algorithm, we are in a situation similar to the one in the convex algorithm

discussed in [25]. The argument presented there can be used and shows convergence to a local minimum $x^\sharp \in \Omega$. □

Remark. The present trust region method should be compared to the approach of Fuduli, Gaudioso, and Giallombardo [20, 21] for general nonsmooth and nonconvex locally Lipschitz functions, where the authors design a trust region with the help of the first order affine approximations $a(y) = g^\top(y - y^{k+1}) + f(y^{k+1})$, $g \in \partial f(y^{k+1})$ of the objective f at the trial points y^{k+1}. As these affine models are not support functions to the objective, the authors classify them according to whether $a(x) > f(x)$ or $a(x) \le f(x)$, using this information to devise a trust region around the current x. Their approach is certainly appealing, because it uses genuine information from the objective f. In contrast, our method uses information from the model $\phi(\cdot; x)$ at the trial points y^{k+1}.

9. Minimizing the H_∞-norm. In this section we extend our algorithm to a larger class of functions which are suprema of an infinite family of maximum eigenvalue functions. The application we have primarily in mind is the H_∞-norm, but the results are applicable to a much larger class.

To introduce our case, we consider a parametrized family of stable linear time-invariant dynamical systems

$$P(\theta) : \begin{cases} \dot{x} = A(\theta)x + B(\theta)w, \\ z = C(\theta)x + D(\theta)w \end{cases} \tag{26}$$

with data $A(\theta) \in \mathbb{R}^{n_x \times n_x}$, $B(\theta) \in \mathbb{R}^{n_w \times n_x}$, $C(\theta) \in \mathbb{R}^{n_x \times n_z}$, $D(\theta) \in \mathbb{R}^{n_w \times n_z}$ depending smoothly on a decision parameter $\theta \in \mathbb{R}^n$. The transfer function of $P(\theta)$ is $G(\theta, s) = C(\theta)\left(sI - A(\theta)\right)^{-1} B(\theta) + D(\theta)$. Here n_x is the order of the system, $x(t)$ its state, n_w the number of inputs, $w(t)$ the input vector, n_z the number of outputs, and $z(t)$ the output vector. As a typical example, in feedback control synthesis, $P(\theta)$ may represent a closed-loop system, depending on the unknown (to be designed) feedback controller θ. The closed-loop transfer function then depends on the decision vector θ, which regroups the controller gains and possibly other decision parameters, e.g., from the open-loop system [44], or scalings/multipliers in robust synthesis [7].

Typically, the performance of the unknown feedback controller might be assessed in the H_∞-norm. Recall that the H_∞-norm $\|G(\theta, \cdot)\|_\infty$ of a stable system is the $L^2(j\mathbb{R}) \to L^2(j\mathbb{R})$ operator norm of the channel $w \to z$, where $z(s) = G(\theta, s)w(s)$. An explicit expression is

$$\|G(\theta, \cdot)\|_\infty = \sup_{\omega \in \mathbb{R} \cup \{\infty\}} \overline{\sigma}\left(G(\theta, j\omega)\right) = \sup_{\omega \in \mathbb{R} \cup \{\infty\}} \lambda_1\left(G(\theta, j\omega)^H G(\theta, j\omega)\right)^{1/2},$$

where X^H is the conjugate transpose of a matrix X. We are interested in that choice of θ which minimizes the H_∞-norm,

$$\min_{\theta \in \mathbb{R}^n} \|G(\theta, \cdot)\|_\infty. \tag{27}$$

We introduce the function

$$f(\theta) = \|G(\theta, \cdot)\|_\infty^2,$$

which is then an infinite maximum of maximum eigenvalue functions

$$f(\theta) = \max_{\omega \in \mathbb{R} \cup \{\infty\}} f(\theta, \omega), \qquad f(\theta, \omega) = \lambda_1\left(F(\theta, \omega)\right),$$

where

$$F(\theta,\omega) = G(\theta,j\omega)^H G(\theta,j\omega) \in \mathbb{S}^m.$$

Program (27) is semi-infinite with two sources of nonsmoothness: the infinite maximum operator and the nonsmoothness of each maximum eigenvalue function $f(\cdot,\omega)$.

Yet another difficulty arises in (27). Namely, given the fact that the H_∞-norm is defined only for stable transfer functions, the objective function $f(\theta)$ is defined only on the set $\mathcal{S}$ of those parameters θ where $G(\theta,\cdot)$ is stable. In other words, program (27) has the hidden constraint $\theta \in \mathcal{S}$. But $\mathcal{S}$ is an open set, because $G(\theta,\cdot)$ depends continuously on θ, so $\theta \in \mathcal{S}$ is not a constraint in the usual sense of mathematical programming. The following known fact is therefore useful.

LEMMA 8. *Suppose $(A(\theta),B(\theta),C(\theta),D(\theta))$ is observable and controllable for every $\theta \in \mathcal{S}$. Then $\|G(\theta,\cdot)\|_\infty \to +\infty$ for $\theta \in \mathcal{S}$ and $\theta \to \bar{\theta} \in \partial\mathcal{S}$. In other words, $f(\theta) = \|G(\theta,\cdot)\|_\infty^2$ behaves like a barrier function as θ approaches the boundary $\partial\mathcal{S}$ of the hidden constraint $\mathcal{S}$.*

The following result is yet another key property for the analysis of f; see, e.g., [11], [10, Lemma 1] for a proof.

LEMMA 9. *Suppose $G(\theta)$ is stable, i.e., $\theta \in \mathcal{S}$. Then the set $\Omega(\theta) = \{\omega \in \mathbb{R}\cup\{\infty\} : f(\theta) = f(\theta,\omega)\}$ of active frequencies is either finite or $\Omega(\theta) = \mathbb{R}\cup\{\infty\}$, i.e., $f(\theta) = f(\theta,\omega)$ for every $\omega \in \mathbb{R}\cup\{\infty\}$.*

We refer to $\Omega(\theta)$ as the set of active frequencies at θ. A system where $\Omega(\theta) = \mathbb{R}\cup\{\infty\}$ is called *all-pass*. In practical cases, iterates θ where $G(\theta,\cdot)$ is all-pass are rarely encountered.

For the following we switch back to the more standard notation in optimization, where the decision variable θ is denoted by $x \in \mathbb{R}^n$. Let x be our current iterate and consider the case where $\Omega(x) = \{\omega_1,\dots,\omega_p\}$ is finite. Any Ω with $\Omega(x) \subset \Omega \subset \mathbb{R}\cup\{\infty\}$ is called an extension of $\Omega(x)$. For a given extension Ω we consider the function $f_\Omega(y) = \max_{\omega\in\Omega} f(y,\omega)$. If Ω is finite, then f_Ω is a maximum eigenvalue function, namely, $f_\Omega(y) = \lambda_1(F_\Omega(y))$, where $F_\Omega(y)$ is block diagonal with diagonal blocks $F(y,\omega)$, $\omega \in \Omega$ arranged in any convenient order. We have $f_\Omega \le f$ and $f_{\Omega(x)}(x) = f_\Omega(x) = f(x)$ for every extension Ω of $\Omega(x)$. The subdifferential of f at x is determined by $\Omega(x)$ in as much as

$$\partial f(x) = \partial f_\Omega(x) = \partial f_{\Omega(x)}(x).$$

Our goal is to extend the eigenvalue optimization algorithm to the case of the H_∞-norm. We use the following simple idea:

i. For a finite extension Ω of $\Omega(x)$ we know how to generate descent steps for f_Ω at x, because f_Ω is a maximum eigenvalue function.
ii. Suppose y^{k+1} is a serious step for f_Ω satisfying the acceptance test in step 4 of the algorithm. If Ω is large enough, f_Ω is close to f, so that we may hope that the acceptance test will also be satisfied for f.

This leads to a convergent algorithm for the H_∞-norm. What is needed is an increasing sequence $\Omega^1 \subset \Omega^2 \subset \cdots$ of finite sets whose union is dense in $\mathbb{R}_\infty$. Then we use the scheme of our algorithm to generate descent steps for f_{Ω_ℓ}, where $\Omega(x)\cup\Omega^\ell \subset \Omega_\ell$. If the approximation of f by f_{Ω_ℓ} is not good enough, we replace Ω_ℓ by the larger $\Omega_{\ell+1}$, where $\Omega(x)\cup\Omega^{\ell+1} \subset \Omega_{\ell+1}$, etc. This approach is inspired by the theory of consistent approximations of [53].

Spectral bundle algorithm for program (27).

Parameters $0 < \gamma^\sharp < \gamma < \frac{1}{2}$.	
0.	**Initialize outer loop.** Choose initial x such that $f(x) < \infty$.
1.	**Outer loop.** If $0 \in \partial f(x)$ at current x stop, else goto inner loop.
2.	**Initialize inner loop.** Let $x_1 = x$ and choose finite Ω_1 containing $\Omega(x_1)$. Put inner loop counter $\ell = 1$.
3.	**Subprogram.** At inner loop counter ℓ and current Ω_ℓ, f_{Ω_ℓ}, $\phi_{\Omega_\ell}(\cdot;x)$, and $\phi^k_{\Omega_\ell}(\cdot;x)$ use inner loop of the first algorithm (with counter k) to generate trial step x_ℓ satisfying the test $\dfrac{f(x)-f_{\Omega_\ell}(x_\ell)}{f(x)-\phi^k_{\Omega_\ell}(x_\ell;x)} \geq \gamma.$
4.	**Reality check.** Test whether $\dfrac{f(x)-f(x_\ell)}{f(x)-\phi^k_{\Omega_\ell}(x_\ell;x)} \geq \gamma^\sharp.$
5.	**Decision.** If this is the case, let $x^+ = x_\ell$ and go back to step 1. Otherwise add new frequencies to the set Ω_ℓ to obtain $\Omega_{\ell+1}$ and go back to step 3.

Here $\phi_{\Omega_\ell}(\cdot;x)$ relates to f_{Ω_ℓ} as $\phi(\cdot;x)$ relates to f in the algorithm of section 6, and, similarly, $\phi^k_{\Omega_\ell}(\cdot;x)$ used here plays the role of $\phi_k(\cdot;x)$ there.

Remarks. (i) Notice that our algorithm now has three iterative levels: the outer loop generating the serious iterates $x, x^+, x^{++}, \ldots$; the inner loop with counter ℓ, which corresponds in fact to the outer loop in the first algorithm, now applied to the function f_{Ω_ℓ}; and the innermost loop, which corresponds to the inner loop in the first algorithm, and which has its own counter k.

(ii) Notice that Ω_ℓ could in principle be *any* increasing sequence of finite sets of frequencies whose union is dense, but it is preferable to adapt this sequence to the local situation at the current iterate x. Ideas of how $\Omega_\ell(x)$ could be chosen at each step are discussed in [4].

THEOREM 10. *Suppose $(A(x), B(x), C(x), D(x))$ is observable and controllable for every $x \in \mathcal{S}$. Let x^1 be a starting point such that $f(x^1) < \infty$ and such that $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is bounded. Suppose the approximating sequence Ω_k is such that $f_{\Omega_k} \to f$ uniformly on bounded sets as $k \to \infty$. Then every accumulation point of the sequence of iterates x^j generated by the above algorithm with starting point x^1 is a critical point of f.*

Proof. (i) Observe first that due to the barrier property of the objective, the boundedness of the initial level set, and the fact that our method is of descent type in the outer iterates, every accumulation point $\bar{x}$ of the sequence of serious iterates x^j is necessarily inside the stability region $\mathcal{S}$. This means criticality of $\bar{x}$ is still described by $0 \in \partial f(\bar{x})$. In other words, the hidden constraint $x \in \mathcal{S}$ can be disregarded in what follows.

(ii) Let x be the current iterate of the outer loop and consider the inner loop with function f_{Ω_ℓ} and its models ϕ_{Ω_ℓ} and $\phi^k_{\Omega_\ell}$ for a fixed set Ω_ℓ. Applying the lemmas of section 7 to the maximum eigenvalue function f_{Ω_ℓ} shows finite termination of step 3 of the semi-infinite algorithm at a suitable x_ℓ (that is, after a finite number of steps y^{k+1}, where k is the counter of the innermost loop). Notice here that Lemmas 4 and 5 do not use compactness of the level sets of the objective, which is good news, because the objective is f_{Ω_ℓ}, and we know nothing about compactness of the level sets of f_{Ω_ℓ}. Only compactness of the level set of f is assumed in the statement. Instead, what made that argument in section 7 work was the special structure (7) of the subgradients

of the maximum eigenvalue function, and this applies to each f_{Ω_ℓ}.

(iii) The trial iterate x_ℓ found in step 3 corresponds in fact to the latest y^{k+1} of the innermost loop in the terminology of section 7, and the test in step 3 is precisely the acceptance test in the first algorithm. But being built on f_{Ω_ℓ}, x_ℓ does not necessarily pass the reality check in step 4, so a restart with a larger $\Omega_{\ell+1}$ may be required. What we have to prove, then, is that after a finite number of such updates $\Omega_\ell \to \Omega_{\ell+1}$, the $x_\ell = y^{k+1}$ will pass the test in step 4 and become the new outer iterate x^+. This is where we have to use the fact that f_{Ω_ℓ} gets closer to f as Ω_ℓ increases. More precisely, exploiting again the special structure of the subgradients of the different λ_1 involved, and using that $F'(z,\omega)$ is uniformly bounded for z in a bounded set and $\omega \in \mathbb{R}\cup\{\infty\}$, we see that the sequence of trial steps x_ℓ is bounded. Since $f_{\Omega_\ell} \to f$ uniformly on bounded sets, we conclude using $\gamma^\sharp < \gamma$ that ultimately the test in step 3 is sharper than the test in step 4. That proves finiteness of the loop in ℓ.

(iv) Finally, relabeling the outer iterates $x, x^+, \dots$ as x^j, we are back in the situation analyzed in section 8. Using compactness of $\{x \in \mathbb{R}^n : f(x) \le f(x^1)\}$, we can use the same argument, now involving the parameter $\gamma^\sharp$ from step 4 of the algorithm. This completes the argument. □

Remark. The theory of consistent approximations [53] allows us in principle to apply this method in a fairly general context. However, a difficulty arises in step 4 of the algorithm, where the reality check requires computing values $f(x_\ell)$. This is what makes the case of the H_∞-norm special, because here we have an efficient way to compute function values [11]. The same idea can be used to solve problems with integral quadratic constraints (IQCs); see [7, 3].

10. Numerical tests. We present numerical tests with BMIs arising in feedback controller synthesis. Consider a closed-loop system of the form (26), where $A(K)$, $B(K), \dots$ depend on the feedback controller K to be designed. The bounded real lemma [12] asserts that the closed-loop transfer channel $w \to z$ has H_∞-norm bounded by γ_∞ if and only if there exists a Lyapunov matrix $X \succ 0$ such that

$$\widetilde{\mathcal{B}}(K,X,\gamma_\infty) = \begin{bmatrix} A(K)^\top X + XA(K) & XB(K) & C(K)^\top \\ B(K)^\top X & -\gamma_\infty I & D(K)^\top \\ C(K) & D(K) & -\gamma_\infty I \end{bmatrix} \prec 0.$$

Fixing a small threshold $\epsilon > 0$, we consider the following nonlinear semidefinite program:

$$\begin{aligned} \text{minimize} \quad & \gamma_\infty \\ \text{subject to} \quad & \widetilde{\mathcal{B}}(K,X,\gamma_\infty) \preceq -\epsilon I, \\ & \epsilon I - X \preceq 0, \end{aligned}$$

which may be solved as an eigenvalue optimization program with decision variable $x = (\mathbf{vec}(K), \mathbf{svec}(X), \gamma_\infty)$ if exact penalization is used. To this end, put $\mathcal{B}(K,X,\gamma_\infty) := \mathrm{diag}[\epsilon I - X; \widetilde{B}(K,X,\gamma_\infty)]$ and fix a penalty parameter $\alpha > 0$ to solve the eigenvalue program $\min_x \lambda_1\left(\gamma_\infty I + \alpha\mathcal{B}(K,X,\gamma_\infty)\right)$. An alternative approach is to fix the performance level γ_∞ and to solve the eigenvalue program

$$\min_{K,X} \lambda_1\left(\mathcal{B}(K,X,\gamma_\infty)\right) \tag{28}$$

until a value < 0 is found. The gain γ_∞ could then be updated a few times to improve performance. This approach has been adopted in our numerical tests, while the exact

penalty approach was used in [8]. Further testing of this approach for control problems with IQCs is presented in [7].

10.1. Numerical implementation. We have performed six numerical experiments using models known in the control literature (VTOL helicopter, chemical reactor, transport airplane, piezoelectric actuator, coupled springs model, and binary distillation tower). This allows comparison with previous studies. We present both static and reduced order controller designs. The state space matrices of these models can be found in [35, 8], with the results of H_2 and H_∞ synthesis problems.

To solve the nonconvex eigenvalue optimization problem (28), we use our MATLAB implementation of the spectral bundle algorithm. The tangent subproblem to compute the trial step y^+ requires minimizing a quadratic cost function subject to an SDP constraint (an LMI (linear matrix inequality)). In order to solve the tangent subproblem efficiently, our specSDP routine [8] was used.

10.1.1. Initialization of the algorithm. The parameter values of the spectral bundle algorithm in section 6 have been set to $\gamma = 0.01$, $\tilde{\gamma} = 0.4$, and $\Gamma = 0.6$. We use $\mathrm{tol} = 10^{-5}$ as a tolerance parameter to stop the algorithm as soon as progress in function values is minor, that is, $f(x) - f(x^+) < 10^{-5}(|f(x)| + 1)$.

Initialization of the variables X and K in program (28) is a difficult task. Indeed, the cost function (28) is nonconvex and the behavior of the algorithm could dramatically change for a bad choice of X and K. For instance, simple initializations such as $K = 0$ and $X = I$ are bound to fail. We have decided to start with a closed-loop stabilizing K, which is easily obtained via minimization of the spectral abscissa $\alpha(K) = \max \mathrm{Re}\Lambda(A(K))$, where $\Lambda(A(K))$ is the spectrum of $A(K)$; see [9]. Once the initial K^0 is fixed in (28), minimizing the cost function with respect to X alone is a convex program, which can be solved using standard LMI techniques. A possible way to initialize X is therefore to choose X^0 optimal with respect to K^0. Unfortunately, this often leads to numerical problems since X^0 can be ill-conditioned and have an exceedingly large norm. We have observed during our numerical testing that the algorithm may crash because of the difference of magnitude between X and K. To avoid these effects, we have sometimes used a scaled version of X^0 to obtain decision variables K^0, X^0 of the same order of magnitude.

Initializing γ_∞ is easier, because the standard full order H_∞ controller gives a lower bound.

A delicate point is the initialization and choice of the number $r_k \in \mathbb{N}$ defining the dimension of the set $\mathcal{G}_k$ used to define the local model ϕ_k at each sweep k. As there does not seem to be any theoretical way to set the value of r_k, we have adjusted it heuristically during the computations. Figure 1 compares, for the helicopter model, static choices $r_k = \mathrm{const}$ and displays the behavior of the algorithm for $r_k \in \{1, 2, 3, 4\}$. The ratio $f_i(x_k)/f_4(x_k)$ is plotted for $i = 1, \ldots, 3$, and $k = 1, \ldots, 100$, where $f_i(x_k)$ is the value of the cost function for $r_k = i$, and for the kth step x_k of the algorithm. As we can see in this plot, after some iterations, the algorithm behaves best for $r_k = 4$, indicating that larger r_k should give better results. The results in [48] seem to indicate that r_k should be chosen in such a way that the gap between λ_{r_k} and λ_{r_k+1} is as large as possible, but our testing in [2, 4, 8] has not confirmed this. The situation is far from clear, and dynamic choices of r_k ought to be tested. The advantage of our present approach motivated by [25] over the line motivated by [48] is that convergence of the method no longer hinges on the choice of r_k, respectively, r_ϵ.

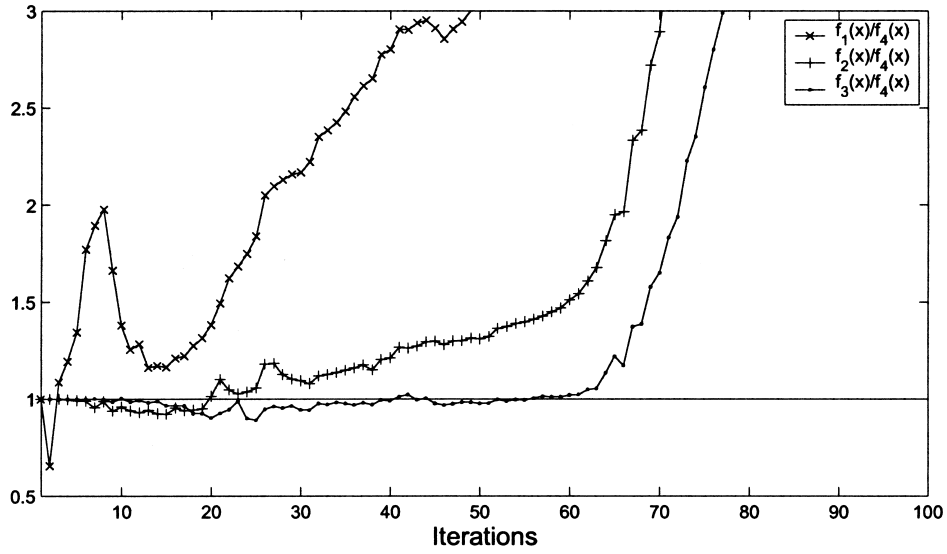

FIG. 1. *Behavior of the cost function during iterations for different values of r_k. This plot shows the ratio $f_i(x_k)/f_4(x_k)$, where $f_i(x_k)$ is the value of the cost function at step x_k for $r_k = i$.*

10.1.2. Nonsmooth optimality tests. To check whether the algorithm has reached a critical point, respectively, a minimum $x^* = (X^*, K^*)$, we have implemented two nonsmooth tests of optimality. The first one uses the ϵ-*enlarged subdifferential* of [48] for the maximum eigenvalue function, $\delta_\epsilon \lambda_1(X)$, to compute the criticality measure

$$\sigma_\epsilon = \operatorname{dist}\left(0, \delta_\epsilon f(x^*)\right),$$

where $f(x) = \lambda_1\left(\mathcal{B}(K, X, \gamma_\infty)\right) = \lambda_1(F(x))$. This parameter is the minimum value of the small size semidefinite program (computed using specSDP [8]):

$$\text{minimize } \left\{||F'(x^*)^\star G||_2 : \quad G \in \delta_\epsilon \lambda_1\left(\mathcal{B}(X^*, K^*, \gamma_\infty)\right)\right\},$$

where $\delta_\epsilon \lambda_1(X) = \left\{Q_\epsilon Y Q_\epsilon^\top \ : \ Y \succeq 0, \operatorname{Tr}(Y) = 1\right\}$. Here Q_ϵ is an $m \times r_\epsilon$ matrix whose r_ϵ columns form an orthonormal basis of eigenvectors associated with those eigenvalues $\lambda_i(X)$ satisfying

$$i \in I_\epsilon := \left\{i \ : \ \lambda_i(X) > \lambda_1(X) - \epsilon\right\}.$$

The number $r_\epsilon := \max\{i \ : \ i \in I_\epsilon\}$ is called the ϵ-*multiplicity* of $\lambda_1(X)$. In [48] it is shown that $\partial\lambda_1(X) \subset \delta_\epsilon\lambda_1(X) \subset \partial_\epsilon\lambda_1(X)$, so that ϵ could be roughly interpreted in the following way: If $0 \in \delta_\epsilon f(x^*) = F'(x^*)^\star \delta_\epsilon \lambda_1\left(\mathcal{B}(X^*, K^*, \gamma_\infty)\right)$, then it is not possible to further decrease f locally around x^* by more than ϵ. See [45, Lemma 2] for more details on this optimality test. Notice that this test may indeed be used as a stopping test in step 1 of the algorithm.

Our second optimality test is heuristic and is designed for a posteriori testing of criticality. It uses random perturbations x of $x^* = (X^*, K^*)$ to see whether the cost function value can be further decreased locally. Denoting by n_v the number of real optimization variables, we generate $100n_v$ random perturbations around $x^* = (X^*, K^*)$. The cost function values $f(X_i, K_i)$ for each perturbation $i = 1, \ldots, 100n_v$

are used to define

$$m_f := \min_i f(X_i, K_i),\ M_f := \max_i f(X_i, K_i),$$

and

$$p_f = \frac{\#\{i : f(X_i, K_i) < f(X^*, K^*)\}}{100 n_v}.$$

Parameter p_f (in percent) gives the proportion of perturbations for which the cost function value is improved.

10.2. Numerical results for H_∞-synthesis. Tables 1 and 2 present results obtained with the spectral bundle algorithm applied to the H_∞-synthesis problem. Table 1 is for static controllers, and Table 2 is for dynamic controllers. Comparison of performance γ_∞ with results in the literature, for static controllers $n_k = 0$, is given in Table 3. Table 4 gives some measured CPU times for three models on the static synthesis case. All numerical experiments have been performed on a Linux computer with a 2Ghz processor.

10.2.1. VTOL helicopter. State space data for the VTOL helicopter model are from [35, 8]; the model is described in [30]. The H_∞ gain was fixed at $\gamma_\infty = 0.1542$, the optimization variables were initialized as $K = [1, 1]^\top$, and $X = I$. The algorithm successfully solved the problem and obtained the H_∞ controller

$$K_\infty = \begin{bmatrix} 14.06432 \\ 239.5975 \end{bmatrix}.$$

We decided to look for a dynamic controller of order $n_k = 2$ with prescribed closed-loop performance $\gamma_\infty = 0.133$. The algorithm was initialized with a closed-loop stabilizing K^0 and $X^0 = I$. The optimal Lyapunov matrix X with respect to the given K^0 was used neither in the static nor in the dynamic case, because it has a very high norm and is likely to introduce numerical problems. The algorithm successfully computed

$$A_K = \begin{bmatrix} 1.672546 & 1.851477 \\ 1.849434 & 1.670218 \end{bmatrix},\ B_K = \begin{bmatrix} 73.76900 \\ 73.68110 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 1.309171 & 1.308932 \\ 3.245753 & 3.241668 \end{bmatrix} \text{e-2},\ D_K = \begin{bmatrix} 0.4300008 \\ 0.7486698 \end{bmatrix}.$$

10.2.2. Chemical reactor. The chemical reactor model and numerical data can be found in [29]. We fixed the performance level $\gamma_\infty = 1.1830$ and initialized our algorithm with a closed-loop stabilizing K^0 together with the associated optimal X^0 (scaled for numerical convenience). This kind of initialization was used for both static and dynamic cases. The obtained static controller is

$$K_\infty = \begin{bmatrix} -3.791707 & -9.704666 \\ -7.166853 & -35.27994 \end{bmatrix}.$$

We also computed a dynamic controller of order 2. For $\gamma_\infty = 1.1420$, we obtained

$$A_K = \begin{bmatrix} -2.197969 & -0.341903 \\ -0.334860 & 2.205355 \end{bmatrix}, \; B_K = \begin{bmatrix} 0.4983757 & 1.096069 \\ -6.452330 & -13.82350 \end{bmatrix},$$

$$C_K = \begin{bmatrix} -0.1090918 & 1.556743 \\ -0.2769249 & 3.964105 \end{bmatrix}, \; D_K = \begin{bmatrix} -3.637238 & -6.382226 \\ -3.506708 & -19.39329 \end{bmatrix}.$$

The criticality measure was quite low in the static and the dynamic case, with $\sigma_\epsilon = 2.0041\text{e-}4$, respectively, $\sigma_\epsilon = 9.7570\text{e-}4$. In the static case, for the purpose of testing, we run the algorithm with a more severe stopping criterion to see if criticality decreased further. The stopping criteria were

$$f(x) - f(x^+) < 10^{-5}(|f(x)| + 1) \text{ and } ||x - x^+|| < 10^{-5}(||x|| + 1). \tag{29}$$

With this rule the algorithm stops after 26448 iterations. The final point verifies $\lambda_1(\widetilde{\mathcal{B}}) = -0.0079$ and criticality measure $\sigma_\epsilon = 4.2865\text{e-}06$, with $r_\epsilon = 4$ and $\epsilon = 1.3813\text{e-}5$. At this numerical precision, we can consider that the algorithm has reached a critical point.

10.2.3. Transport airplane. Model and state space data for the transport airplane are from [22]. We used a closed-loop stabilizing K^0 and the associated optimal X^0 for initialization. For $\gamma_\infty = 3.1770$, our algorithm computed the static controller

$$K_\infty = \begin{bmatrix} 0.6340988 & -0.5964908 & -0.7923650 & 5.166775e-2 & 1.055142 \end{bmatrix}.$$

In the static case we have also made a test of the algorithm with the stopping criterion (29). We have observed that the criticality of the final point of the algorithm has decreased: $\sigma_0 = 2.4819\text{e-}5$ after 241 iterations. Again, we can consider that the algorithm has reached a critical point with regard to the chosen numerical precision.

We failed to find a dynamic controller of order 2 for the airplane model. We computed a dynamic controller of order 1 with performance $\gamma_\infty = 2.860$. The H_∞ controller is

$$A_K = \begin{bmatrix} -0.4589498 \end{bmatrix},$$

$$B_K = \begin{bmatrix} -1.133331 & 1.441023 & 1.107071 & -0.116483 & -1.873279 \end{bmatrix} \text{e-}2,$$

$$C_K = \begin{bmatrix} -7.108071\text{e-}3 \end{bmatrix},$$

$$D_K = \begin{bmatrix} 1.740566 & -0.8878559 & -0.9477933 & 0.1000800 & 2.542508 \end{bmatrix}.$$

10.2.4. Piezoelectric actuator. The model of the piezoelectric actuator can be found in [13]. This study turned out to be one of the most difficult. As can be seen in Tables 1 and 2, the spectral bundle algorithm at first failed to solve the control problem both in the static and in the dynamic case. The algorithm converged to a couple (X^*, K^*) with slightly positive objective value around 1.45e-5 for the dynamical case, with the criticality parameter σ_ϵ quite small in both cases, indicating optimality. While it is perfectly possible that our algorithm, which is a local optimization method, may converge to a local minimum with positive values, failing to solve the underlying control problem, the present case turned out to be special. Namely, upon testing

the obtained controller K^*, we realized that it *is* closed-loop stabilizing and therefore solves the control problem. What happened is that the X^* computed by our algorithm was not suitable, as it fails to solve the Lyapunov inequality. Nonetheless, within the prescribed numerical precision both stopping tests indicated a local minimum.

The static controller for $\gamma_\infty = 0.0578$ reads

$$K_\infty = \left[\begin{array}{ccc} -0.3880673 & -1.837152 & -10.00377 \end{array} \right] \text{e+7},$$

and the dynamical controller of order 2, for $\gamma_\infty = 0.030$, is

$$A_K = \left[\begin{array}{cc} -6.770630 & -7.974432\text{e-}1 \\ -7.973471\text{e-}1 & -5.118056 \end{array} \right] \text{e+6},$$

$$B_K = \left[\begin{array}{ccc} 8.842279\text{e-}2 & 4.722429\text{e-}1 & 2.812368 \\ 3.574044\text{e-}2 & 1.910070\text{e-}1 & 1.136064 \end{array} \right] \text{e+7},$$

$$C_K = \left[\begin{array}{cc} -1.167814 & -4.720797\text{e-}1 \end{array} \right] \text{e+6},$$

$$D_K = \left[\begin{array}{ccc} -4.930334\text{e-}1 & -1.977292 & -1.480471\text{e+}1 \end{array} \right] \text{e+7}.$$

The described phenomenon indicates the numerical difficulty of synthesis problems with joint variable $x = (X, K)$, where important disparities between the numerical ranges of the two variables K and X may occur. In particular, the Lyapunov matrix X may be very ill-conditioned. The idea to keep K^* and compute a new Lyapunov variable X associated with K^* using a convex technique is systematically used in D-K iterations, where K and X variables are optimized alternatingly. The advantage of this approach is that both subproblems are then convex and can be solved by standard SDP solvers. However, intensive testing [27, 18] has shown that D-K techniques tend to get stalled and should in general be avoided. We believe that joint minimization in $x = (X, K)$ is the method of choice, despite the indicated difficulties. This does not exclude occasional restarts.

10.3. Coupled springs model. This is model CSE1 from [35] and consists of a string of coupled springs with dash-pots and masses. Input forces act on both the left and on the right ends of the spring system. The feedback controller has to stabilize the positions of the masses. We focus on the synthesis of a dynamic controller of order 4. To begin with, an initial closed-loop stabilizing controller was computed. Then X^0 was set to identity. An optimal controller was then synthesized for the performance level $\gamma_\infty = 0.0235$. The algorithm stopped at (K^*, X^*) with criticality measure sufficiently low in comparison with the numerical precision: $\sigma_\epsilon = 4.36e-6$, with $r_\epsilon = 5$ and $\epsilon = 3.64e-5$.

10.4. Distillation tower. Finally, to test the efficiency of the algorithm on a larger model, we used the BDT2 model from COMPL$_e$IB library [35], a binary distillation tower with 82 states, 4 outputs, and 4 controller inputs. The complete model is described in [56, section 12.4]. As can be seen in Table 1, the number of optimization variables $n_v = 3419$ is large in comparison with the previous examples. It should be highlighted that approximatively 99.5% (3403) of these variables are needed for the Lyapunov matrix X^*, but only 0.5% (16) are needed for controller K^*.

TABLE 1
Results of static H_∞-synthesis.

	γ_∞	α	$\lambda_n(X)$	$\lambda_1(\widetilde{\mathcal{B}})$	$\lambda_{n+n_y+n_u}(\widetilde{\mathcal{B}})$
Helicopter	0.1542	-0.1288	8.3236	-1.8146e-4	-1.8866e+5
Chemical reactor	1.1830	-1.8721	0.4327	-4.4606e-3	-2.4616e+3
Airplane	3.1770	-0.2486	1.1127e-3	-5.8368e-5	-44.4910
Piezo actuator	0.0578	-1.4202	7.0198e-10	8.2255e-3	-399.3432
BDT2	1.0722	-8.7144e-2	0.3223	-4.3202e-3	-6.2023e+2

#It.	$[n, n_y, n_u]$	n_v	r_k	σ_ϵ	r_ϵ	ϵ	p_f	m_f	M_f
600	[4,2,1]	12	4	1.15e-4	3	3.77e-7	0%	6.43e-4	1.95e-2
1800	[4,2,2]	14	4	2.01e-4	4	1.23e-5	0%	6.63e-3	9.28e-2
77	[9,1,5]	50	12	0.0114	4	3.24e-5	0%	4.00e-7	4.38e-5
1191	[5,1,3]	18	10	7.05e-3	5	7.94e-4	0%	570	2.29e+4
2934	[82,4,4]	3419	15	1.27e-3	6	6.20e-5	0.3%	-1.02e-5	8.87e-4

TABLE 2
Results of dynamic H_∞-synthesis.

	γ_∞	α	$\lambda_n(X)$	$\lambda_1(\widetilde{\mathcal{B}})$	$\lambda_{n+n_y+n_u}(\widetilde{\mathcal{B}})$
Helicopter	0.1334	-0.1583	2.7722e-3	-5.7903e-5	-39.5053
Chemical reactor	1.1420	-0.8245	0.2944	-3.2326e-3	-735.4831
Airplane	2.8600	-0.3161	6.2022e-4	-1.7815e-4	-150.1174
Piezo actuator	0.0300	-0.4796	-9.7169e-6	1.4503e-5	-16.1523
CSE1	0.0235	-2.1309e-1	8.2865e-1	-1.5574e-4	-6.6618e+2

#It.	$[n, n_k, n_y, n_u]$	n_v	r_k	σ_ϵ	r_ϵ	ϵ	p_f	m_f	M_f
9334	[6,2,4,3]	33	10	5.14e-4	5	1.11e-5	0%	4.01e-2	5.20
1379	[6,2,4,4]	37	6	9.75e-4	3	2.94e-4	1.81%	-1.81e-9	2.80e-8
1781	[10,1,2,6]	67	12	4.26e-2	3	4.50e-5	0%	2.11e-3	2.25e-2
8962	[6,2,3,5]	43	10	3.27e-2	5	1.45e-5	0%	1.15e-3	7.44e-2
2e4	[20,4,10,2]	384	6	4.34e-6	5	3.64e-5	0%	0.0025	6.51

TABLE 3
Comparison of γ_∞ with results in the literature for static controllers $n_k = 0$. The nonconvex spectral bundle method (NSBM) is shown on left.

	NSBM	[34]	[8]
Helicopter	0.1542	0.3455	0.157
Chemical reactor	1.1420	-	1.202
Airplane	3.1770	3.1774	2,220
Piezo actuator	0.0578	6.6256	3.055e-3

TABLE 4
Comparison of mean CPU times, in seconds, on three models of different sizes. Mean CPU times are given for computation of $f(x)$, computation of $F'(x)$, resolution of tangent program TP, serious step, and null step. %Serious gives the percentage of serious steps with respect to the total number of iterations.

Mean time	$f(x)$	$F'(x)$	TP	Serious	Null	%Serious
Helicopter	3.23e-04	8.80e-04	3.75e-02	3.81e-02	3.89e-02	35.2
Airplane	5.11e-04	1.46e-03	8.64e-02	8.80e-02	9.09e-02	66.8
BDT2	1.08e-02	1.12	4.45	5.51	4.66	44.5

Synthesis of a static controller K_∞ was obtained for the performance level $\gamma_\infty = 1.0722$. An initial stabilizing controller K^0 was computed with its corresponding optimal Lyapunov matrix X^0. The synthesized controller is

$$K_\infty := \begin{bmatrix} 5.748949e-1 & 1.751953 & 9.954549e-1 & 3.725248e-1 \\ -6.313297e-1 & 1.133587 & 9.815346e-1 & 2.909215 \\ 1.986992 & 1.789245 & 3.988785e-1 & 2.468048 \\ -1.061248e-1 & 5.597463e-1 & 3.635867 & 4.772583 \end{bmatrix}.$$

Criticality was fairly low compared to the numerical precision: $\sigma_\epsilon = 1.27e-3$ with $r_\epsilon = 6$ and $\epsilon = 6.20e-5$. However, the same numerical phenomenon as in the piezoelectric actuator example was observed: (X^*, K^*) was not a stationary point, and the cost could be further reduced by using a convex optimization technique to compute a new Lyapunov matrix X with K^* fixed.

Acknowledgment. The authors would like to thank the anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] P. Apkarian and D. Noll, *Nonsmooth optimization for multidisk H_∞ synthesis*, European J. Control, 12 (2006), pp. 229–244.

[2] P. Apkarian and D. Noll, *Controller design via nonsmooth multidirectional search*, SIAM J. Control Optim., 44 (2006), pp. 1923–1949.

[3] P. Apkarian and D. Noll, *IQC analysis and synthesis via nonsmooth optimization*, Systems Control Lett., 55 (2006), pp. 971–981.

[4] P. Apkarian and D. Noll, *Nonsmooth H_∞ synthesis*, IEEE Trans. Automat. Control, 51 (2006), pp. 71–86.

[5] P. Apkarian and D. Noll, *Nonsmooth optimization for multiband frequency domain control design*, Automatica, 43 (2007), pp. 724–731.

[6] P. Apkarian and D. Noll, *Nonsmooth structured control design with applications to PID loopshaping of a process*, Internat. J. Robust Nonlinear Control, 17 (2007), pp. 1320–1342.

[7] P. Apkarian, D. Noll, and O. Prot, *Nonsmooth methods for analysis and synthesis with integral quadratic constraints*, submitted.

[8] P. Apkarian, D. Noll, J.-B. Thevenet, and H. D. Tuan, *A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis*, Eur. J. Control, 10 (2004), pp. 527–538.

[9] V. Bompart, P. Apkarian, and D. Noll, *Non-smooth techniques for stabilizing linear systems*, in Proceedings of the 2007 American Control Conference (New York, NY), IEEE Press, Piscataway, NJ, 2007, pp. 1245–1250.

[10] V. Bompart, D. Noll, and P. Apkarian, *Second-order nonsmooth optimization of the H_∞ norm*, Numer. Math., 107 (2007), pp. 433–454.

[11] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm*, Systems Control Lett., 15 (1990), pp. 1–7.

[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.

[13] B. M. Chen, *H_∞ Control and its Applications*, Lecture Notes in Control and Inform. Sci. 235, Springer-Verlag, New York, Heidelberg, Berlin, 1998.

[14] F. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Math. Society Series, John Wiley & Sons, New York, 1983.

[15] S. Cox and R. Lipton, *Extremal eigenvalue problems for two-phase conductors*, Arch. Ration. Mech. Anal., 136 (1996), pp. 101–117.

[16] J. Cullum, W. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Stud., 3 (1975), pp. 35–55.

[17] A. R. Diaaz and N. Kikudu, *Solution to shape and topology eigenvalue optimization problems using a homogenization method*, J. Numer. Methods Engrg., 35 (2005), pp. 1487–1502.

[18] B. Fares, D. Noll, and P. Apkarian, *Robust control via sequential semidefinite programming*, SIAM J. Control Optim., 40 (2002), pp. 1791–1820.

[19] R. Fletcher, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.
[20] A. Fuduli, M. Gaudioso, and G. Giallombardo, *A DC piecewise affine model and a bundling technique in nonconvex nonsmooth optimization*, Optim. Methods Softw., 19 (2004), pp. 89–102.
[21] A. Fuduli, M. Gaudioso, and G. Giallombardo, *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control*, SIAM J. Optim., 14 (2004), pp. 743–756.
[22] D. Gangsaas, K. Bruce, J. Blight, and U.-L. Ly, *Applications of modern synthesis to aircraft control: Three case studies*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 995–1014.
[23] C. Helmberg and K. C. Kiwiel, *A spectral bundle method with bounds*, Math. Programming, 93 (2002), pp. 173–194.
[24] C. Helmberg and F. Oustry, *Bundle methods to minimize the maximum eigenvalue function*, in Handbook of Semidefinite Programming. Theory, Algorithms and Applications, Internat. Ser. Oper. Res. Management Sci. 27, L. Vandenberghe, R. Saigal, and H. Wolkowitz, eds., Kluwer, Boston, 2000, pp. 307–337.
[25] C. Helmberg and F. Rendl, *Spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.
[26] C. Helmberg, F. Rendl, and R. Weismantel, *A semidefinite programming approach to the quadratic knapsack problem*, J. Comb. Optim., 4 (2000), pp. 197–215.
[27] J. W. Helton and O. Merino, *Coordinate optimization for bi-convex matrix inequalities*, in Proceedings of the 45th IEEE Conference on Decision and Control (San Diego, CA), IEEE Press, Piscataway, NJ, 1997, pp. 3609–3613.
[28] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Vols. I, II, Grundlehren Math. Wiss. 305, 306, Springer-Verlag, New York, Heidelberg, Berlin, 1993.
[29] Y. S. Hung and A. G. J. MacFarlane, *Multivariable Feedback: A Quasi-classical Approach*, Lecture Notes in Control and Inform. Sci. 40, Springer-Verlag, New York, Heidelberg, Berlin, 1982.
[30] L. H. Keel, S. P. Bhattacharyya, and J. W. Howe, *Robust control with structured perturbations*, IEEE Trans Automat. Control, 36 (1988), pp. 68–77.
[31] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Math. 1133, Springer-Verlag, Berlin, 1985.
[32] K. C. Kiwiel, *A linearization algorithm for computing control systems subject to singular value inequalities*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 595–602.
[33] K. C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable optimization*, Math. Programming, 46 (1990), pp. 105–122.
[34] F. Leibfritz, *Computational Design of Stabilizing Static Output Feedback Controllers*, Tech. Report 99-01, Universität Trier, Trier, Germany, 1999.
[35] F. Leibfritz, *$COMPL_eIB$, COnstrained Matrix-optimization Problem LIbrary—A Collection of Test Examples for Nonlinear Semidefinite Programs, Control System Design and Related Problems*, Tech. Report, Universität Trier, Trier, Germany, 2003.
[36] C. Lemaréchal, *An extension of Davidson's method to nondifferentiable problems*, in Nondifferentiable Optimization, Math. Programming Stud., M. L. Balinski and P. Wolfe, eds., North–Holland, Amsterdam, 1975, pp. 95–109.
[37] C. Lemaréchal, *Bundle methods in nonsmooth optimization*, in Nonsmooth Optimization, Proceedings of the IIASA Workshop (1977), C. Lemaréchal and R. Mifflin, eds., Pergamon, Oxford, Elmsford, NY, 1978, pp. 79–102.
[38] C. Lemaréchal, *Nondifferentiable optimization*, in Optimization, Handbooks Oper. Res. Management Sci. 1, North–Holland, Amsterdam, 1989, pp. 529-572.
[39] C. Lemaréchal, *Lagrangian relaxation*, in Computational Combinatorial Optimization, M. Junger and D. Naddef, eds., Springer, New York, 2001, pp. 112–156.
[40] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, *New variants of bundle methods*, Math. Programming, 69 (1995), pp. 111–147.
[41] C. Lemaréchal and F. Oustry, *Nonsmooth algorithms to solve semidefinite programs*, in Advances in Linear Matrix Inequality Methods in Control, L. El Ghaoui and S.-I. Niculescu, eds., SIAM, Philadalphia, 1999, pp. 57–77.
[42] D. Mayne and E. Polak, *Algorithms for the design of control systems subject to singular value inequalities*, Math. Programming Stud., 18 (1982), pp. 112–134.
[43] D. Mayne, E. Polak, and A. Sangiovanni, *Computer aided design via optimization*, Automatica, 18 (1982), pp. 147–154.

[44] K. Mombaur, *Stability Optimization of Open-Loop Controlled Walking Robots*, Ph.D. Thesis, Universität Heidelberg, Heidelberg, Germany, 2001.
[45] D. Noll and P. Apkarian, *Spectral bundle method for nonconvex maximum eigenvalue functions: First-order methods*, Math. Programming Ser. B, 104 (2005), pp. 701–727.
[46] D. Noll and P. Apkarian, *Spectral bundle method for nonconvex maximum eigenvalue functions: Second-order methods*, Math. Programming Ser. B, 104 (2005), pp. 729–747.
[47] D. Noll, M. Torki, and P. Apkarian, *Partially augmented Lagrangian method for matrix inequality constraints*, SIAM J. Optim., 15 (2004), pp. 161–184.
[48] F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function*, Math. Programming Ser. A, 89 (2000), pp. 1–33.
[49] M. L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.
[50] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.
[51] M. L. Overton and R. S. Womersley, *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 41–45.
[52] E. Polak, *On the mathematical foundations of nondifferentiable optimization in engineering design*, SIAM Rev., 29 (1987), pp. 21–89.
[53] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Appl. Math. Sci. 124, Springer, New York, 1997.
[54] E. Polak and Y. Wardi, *A nondifferential optimization algorithm for the design of control systems subject to singular value inequalities over the frequency range*, Automatica, 18 (1982), pp. 267–283.
[55] F. Pukelsheim, *Optimal Design of Experiments*, John Wiley & Sons, New York, 1993.
[56] S. Skogestad and I. Postlethwaite, *Multivariate Feedback Control*, John Wiley & Sons, New York, 1996.
[57] J.-B. Thevenet, D. Noll, and P. Apkarian, *Nonlinear spectral SDP method for BMI-constrained problems: Applications to control design*, in Informatics in Control, Automation and Robotics I, J. Braz, H. Arajo, A. Viera, and B. Encarnaco, eds., Springer-Verlag, Berlin, 2006, pp. 61–72.
[58] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.
[59] P. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable functions*, in Nondifferentiable Optimization, Math. Programming Stud. 3, M. L. Balinski and P. Wolfe, eds., North–Holland, Amsterdam, 1975, pp. 145–173.
[60] Z. Zhao, B. Braams, M. Fukuda, M. Overton, and J. Percus, *The reduced density matrix method for electronic structure calculations and the role of three-index representability conditions*, J. Chem. Phys., 120 (2004), pp. 2095–2104.