Mathematical Programming manuscript No.
(will be inserted by the editor)

Dominikus Noll · Pierre Apkarian

Spectral bundle methods for non-convex maximum eigenvalue functions: second-order methods

Dedicated to R. T. Rockafellar on the occasion of his 70th anniversary

Received: date / Revised version: date

Abstract. We study constrained and unconstrained optimization programs for nonconvex maximum eigenvalue functions. We show how second order techniques may be introduced as soon as it is possible to reliably guess the multiplicity of the maximum eigenvalue at a limit point. We examine in which way standard and projected Newton steps may be combined with a nonsmooth first-order method to obtain a globally convergent algorithm with a fair chance to local superlinear or quadratic convergence.

Key words. Eigenvalue optimization, first and second-order spectral bundle method, ϵ-subgradients, superlinear and quadratic convergence, bilinear matrix inequality (BMI), linear matrix inequality (LMI), semidefinite programming (SDP).

1. Introduction

This paper continues the study of eigenvalue optimization programs initiated in [31]. We investigate programs featuring nonconvex maximum eigenvalue functions, like the unconstrained eigenvalue program

$$\text{minimize} \qquad f(x) = \lambda_1\left(\mathcal{F}(x)\right), x \in \mathbb{R}^n \tag{1}$$

and the constrained eigenvalue program

$$\begin{array}{l} \text{minimize} \;\; c^\top x, x \in \mathbb{R}^n \\ \text{subject to } f(x) = \lambda_1\left(\mathcal{F}(x)\right) \leq 0 \end{array} \tag{2}$$

Here $\mathcal{F} : \mathbb{R}^n \to \mathbb{S}^m$ is a class $\mathcal{C}^2$ operator with values in the space $\mathbb{S}^m$ of symmetric $m \times m$ matrices, and $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is the maximum eigenvalue function, which is convex but generally nonsmooth. In consequence, f is in general neither smooth nor convex. We investigate in which way first-order spectral bundle techniques for programs (1) and (2) developed in [10,39,31] can be combined with second-order steps in order to get fast local convergence. This is of vital importance

Dominikus Noll: Université Paul Sabatier, Institut de Mathématiques de Toulouse, 118, route de Narbonne, 31062 Toulouse, France

Pierre Apkarian: CERT-ONERA, Control System Department, 2, avenue Edouard Bélin, 31055 Toulouse, France

in practice, since first-order methods have a tendency to get stalled toward the end of the optimization process. In smooth programming, this phenomenon is addressed by the use of second-order techniques, which lead to superlinear or quadratic convergence as soon as iterates get close enough to a local solution. It is a long standing research issue to identify similar second-order methods for nonsmooth programs. Presently we contribute two such methods for eigenvalue optimization, and we discuss and clarify previous approaches in [10,39].

Local second-order methods for eigenvalue optimization have been examined before. Much pioneering work has been contributed by M. Overton in a series of papers [32–36] beginning in the 1980s, where Newton type methods for (1) are considered. Further to be mentioned among the earliest contributions are J. Cullum *et al.* [10], R. Fletcher [12], A. Shapiro [44,45] and A. Shapiro and M.K.H. Fan [46].

The specificity of our contribution is that we combine first and second-order techniques in a unified framework. To our knowledge, the only source where this has been explicitly proposed before is Oustry [39]; see also [18,28], where such a combined strategy has been examined for program (1) with a convex objective $f = \lambda_1 \circ \mathcal{A}$.

The combined approach requires two elements, a first-order method, generating a descent or Cauchy step, x^C, and the second-order technique, proposing a Newton type step x^N. The Cauchy point x^C could for instance be obtained by a spectral bundle algorithm like the one analyzed in [31]. This algorithm is of ϵ-subgradient type and was originally proposed by Cullum *et al.* [10] and further developed by Oustry [39] for convex programs (1). But other techniques could be used instead, like for instance [40,41,2,1,3], where modifications of the ϵ-subgradient strategy are considered.

Second-order methods generating Newton steps x^N are presently analyzed. We examine the corresponding tangent quadratic programs in detail and show that it may be profitable to use a projected Newton or quasi-Newton method, where the partial smoothness of the maximum eigenvalue function along certain manifolds is exploited. For related ideas we refer to Hare and Lewis [17], where the idea of semi-smoothness is developed, and to Mifflin and Sagastizábal [30], where UV-analysis is used to generate second-order steps.

General purpose bundle methods are discussed in Lemaréchal [25,26], Wolfe [48] or Kiwiel [22,23]. Combining those with second-order steps may lead to complications as soon as the bundle procedure builds up memory from previous steps. The subtle mechanism which usually combines null steps and serious steps will have to be modified in order to accommodate steps x^N proposed by the second-order method. Using the memoryless spectral bundle method [10,39, 31] avoids this fallacy. We mention, however, that first and second-order techniques of our combined scheme are modulable in the sense that *any* first-order nonsmooth technique producing the Cauchy step x^C with a convergence certificate could be used within the combined framework, as soon as the mentioned difficulties are dealt with. For instance, in [41], Polak and Wardi present an alternative approach, which could also be adapted to compute a Cauchy step x^C. Yet another possibility is proposed in [1], where this idea is further developed to include semi-infinite cases like the H_∞-norm. We also mention Fletcher [13, Ch. 14], where composite optimization programs of the from $\min_{x\in\mathbb{R}^n} \phi(\mathcal{F}(x))$ with smooth $\mathcal{F}$ and nonsmooth convex ϕ are discussed. The author obtains a second order method if ϕ is a polyhedral function. For non-polyhedral ϕ like λ_1, it would again become necessary to build up a polyhedral approximation of ϕ of increasing complexity.

The outline of the paper is as follows. The first-order method from [31] is briefly recalled in Section 3. Section 4 examines the structure of the second-order tangent programs, while the idea of the projected Newton method is discussed in Sections 5 and 6. The affine case is discussed in Section 7, and a link with the method in [39] is established. Section 8 presents an extension of the Dennis-Moré theorem to projected quasi-Newton methods. The constrained eigenvalue program is discussed at some detail in Sections 9 and 10.

2. Notation

We follow [19] and [8] for notions from convexity and nonsmooth analysis. We consider the Euclidean space $\mathbb{S}^m$ of symmetric $m \times m$ matrices, equipped with the scalar product $X \bullet Y = \mathrm{tr}(XY)$. The differential of an operator $\mathcal{F} : \mathbb{R}^n \to \mathbb{S}^m$ is denoted as $\mathcal{F}'(x)$. Its adjoint $\mathcal{F}'(x)^\star$ is a linear operator $\mathbb{S}^m \to \mathbb{R}^n$. For an affine $\mathcal{A} : \mathbb{R}^n \to \mathbb{S}^m$, $\mathcal{A}(x) = A_0 + \sum_{i=1}^n x_i A_i$, the derivative $\mathcal{A}'$ is simply the linear part A of $\mathcal{A}$, given as $Ax = \sum_{i=1}^n x_i A_i$. Its adjoint is then $A^\star Z = (A_1 \bullet Z, \ldots, A_n \bullet Z)$.

3. The ϵ-management

The spectral bundle method analyzed in [31] was originally developed by Cullum *et al.* [10] and Oustry [39] for convex $f = \lambda_1 \circ \mathcal{A}$. It is based on the the following mechanism. Let $x \in \mathbb{R}^n$ be the current iterate, and put $X = \mathcal{F}(x) \in \mathbb{S}^m$. Choose $\epsilon > 0$ and keep the indices $i = 1, \ldots, r(\epsilon)$ of those eigenvalues $\lambda_i(X)$ of X satisfying $\lambda_1(X) \geq \lambda_i(X) > \lambda_1(X) - \epsilon$. Here $r(\epsilon)$ is called the ϵ-multiplicity of $\lambda_1(X)$. Now choose a matrix Q_ϵ of size $r(\epsilon) \times m$, whose columns form an orthonormal basis of the invariant subspace of X associated with the first $r(\epsilon)$ eigenvalues. Then define

$$\delta_\epsilon f(x) = \left\{ \mathcal{F}'(x)^\star G : G = Q_\epsilon Y Q_\epsilon^\top, Y \succeq 0, \mathrm{tr}(Y) = 1, Y \in \mathbb{S}^{r(\epsilon)} \right\},$$

called the ϵ-enlarged subdifferential of f at x. This set satisfies $\partial f(x) \subset \delta_\epsilon f(x) \subset \partial f_\epsilon(x)$, as proved in [39], and serves as a good inner approximation of $\partial_\epsilon f(x)$.

As is well-known, the force of the ϵ-subdifferential for convex functions f is based on the fact that $0 \notin \partial f_\epsilon(x)$ allows to decrease the value of the objective function by at least $\epsilon > 0$. This basic mechanism has to be refined if the approximation $\delta_\epsilon f$ of $\partial_\epsilon f$ is used, as presented in Oustry [39]. If $f = \lambda_1 \circ \mathcal{F}$ is no longer convex, the ϵ-subdifferential and also $\delta_\epsilon f$ loose their global properties, which makes quantifying descent a more complicated task. This is studied in the first part [31] of this work.

Choosing $\epsilon > 0$ to generate a suitable descent step for f at x is referred to as the ϵ-management (see [39], [31]). In the essence, ϵ must meet the following two criteria. We need $0 \notin \delta_\epsilon f(x)$. This ensures that the direction d of steepest ϵ-enlarged descent, obtained by solving the semidefinite program

$$d = -\frac{g}{\|g\|}, \qquad g = \operatorname{argmin}\{\|g\| : g \in \delta_\epsilon f(x)\}$$

is a descent direction of f at x. Secondly, if possible, the choice of ϵ should give the most sizable descent away from x. In particular, descent should be quantifiable, and should be realizable by a finite line search. How all this should be organized is shown in [31] and also [1,3], where variations of the same theme are considered. During the following, the first-order descent step in question will be denoted by x^C, and will be referred to as the Cauchy step. In the smooth case, it corresponds to a standard steepest descent step away from the current x.

4. Second-order methods

Second-order techniques with local superlinear or quadratic convergence are brought into play in eigenvalue optimization with the help of an *oracle* predicting the multiplicity $\bar{r}$ of λ_1 at the limit $\bar{X} = \mathcal{F}(\bar{x})$ during the terminal phase of the algorithm. Naturally, there cannot be any *rigorous* way of forecasting $\bar{r}$, hence the name of an oracle. But heuristic methods have been proposed in the literature and work quite well in practice (see e.g. [12,32–34]). For instance, Overton [33, Section 4] suggests the following estimate r of $\bar{r}$. Choose the smallest r such that

$$\begin{aligned} \lambda_1(X) - \lambda_r(X) &\leq \tau \max\left(1, |\lambda_1(X)|\right), \\ \lambda_1(X) - \lambda_{r+1}(X) &> \tau \max\left(1, |\lambda_1(X)|\right), \end{aligned} \tag{3}$$

where $\tau > 0$ is some small tolerance, which has to be adjusted a few times during the course of the minimization process.

Guessing the multiplicity of $\lambda_1(\bar{X})$ while visiting iterates $X_k = \mathcal{F}(x_k)$ near $\bar{X} = \mathcal{F}(\bar{x})$ may be expressed in terms of the ϵ-management. We aim at $r(\epsilon_k) = \bar{r}$ when x_k is close to $\bar{x}$. As long as this works out successfully, our method will indeed guarantee fast local convergence. On the other hand, if we fail to identify $\bar{r}$, our local quadratic model will be incorrect. The risk we are then taking is that our higher computational load, needed to generate second-order steps, is wasted in so far as convergence is no better than that of the underlying first-order method. But we do not put convergence itself at stake as long as we rely on first-order information based on the true objective function f. The overall scheme is outlined in Figure 1.

The basic idea of the scheme in Figure 1 is clear. The Cauchy step x^C gives a convergence certificate in the sense elaborated in [31] or [1,3]. The test in

Combined algorithm for (1)

1.	If $0 \in \partial f(x)$ stop.
2.	At current x, compute Cauchy point x^C using the first order spectral bundle method [31]. Compute progress $p^C = f(x) - f(x^C) > 0$.
3.	Using (3), make a guess r of the unknown multiplicity $\bar{r}$ of the leading eigenvalue λ_1 in the limit. Pick $\epsilon > 0$ such that $r = r(\epsilon)$.
4.	Form a second-order model based on $r(\epsilon)$ and compute a Newton step x^N by solving the tangent quadratic program, using line search, trust regions or a filter.
5.	Compute decrease $p^N = f(x) - f(x^N)$.
6.	Accept x^N as new iterate x^+ if $p^N \geq \theta p^C$ for fixed $0 < \theta < 1$. Otherwise let $x^+ = x^C$.
7.	Replace x by x^+ and go back to step 1.

Fig. 1.

step 6 assures that if the second-order trial step x^N fails from various reasons, (progress too small: $p^N < \theta p^C$, or even no progress at all: $p^N \leq 0$), we can at least fall back to the progress $p^C > 0$ achieved by x^C, take this as the new iterate, and proceed. On the other hand, if the information from the oracle was sound, and if the Newton step x^N takes its grip, then we have a fair chance to assure superlinear or quadratic convergence toward $\bar{x}$. To our knowledge, the only reference where such a combined scheme was proposed is Oustry [39], who treats the case of a convex $f = \lambda_1 \circ \mathcal{A}$.

Naturally, the scheme in Figure 1 leaves various questions to be resolved. First of all, we have to clarify in which way the Newton step x^N should be computed. Propositions will be made in sections 5, 6 and 10. Secondly, if we decide for instance to use a trust region strategy to compute x^N, we have the problem that we cannot match the values of the tangent quadratic model with the values of $f = \lambda_1 \circ \mathcal{F}$, as we would naturally do in the smooth case. This is because our model relies on the guess r, so if $r \neq \bar{r}$, the local quadratic model does not represent f correctly. In consequence, the usual updating strategy for the trust region radius is not guaranteed to terminate finitely. Similar comments apply to a line search strategy.

Another more technical question is whether elements needed to compute the first-order step x^C may be recycled to build second-order elements, and vice versa. Before we settle these and other problems, we will have to elaborate the precise form of the tangent program based on knowledge of r.

5. Merits of the oracle

Suppose at iterate x_k we have correctly guessed the limiting multiplicity $\bar{r}$ and chosen ϵ_k so that $r(\epsilon_k) = \bar{r}$. This means that at the local minimum $\bar{x}$, the matrix

$\bar{X} = \mathcal{F}(\bar{x})$ lies in the smooth manifold

$$\mathcal{M}_{\bar{r}} = \{X \in \mathbb{S}^m : \lambda_1(X) = \cdots = \lambda_{\bar{r}}(X) > \lambda_{\bar{r}+1}(X) \geq \cdots \geq \lambda_m(X)\},$$

whose dimension is $\frac{m(m+1)}{2} + 1 - \frac{\bar{r}(\bar{r}+1)}{2}$ (cf. [46,21]). Now observe that on $\mathcal{M}_{\bar{r}}$, the maximum eigenvalue λ_1 coincides with the average of the first $\bar{r}$ eigenvalues:

$$\hat{\lambda}_{\bar{r}}(X) := \frac{1}{\bar{r}} \sum_{i=1}^{\bar{r}} \lambda_i(X),$$

which is a convex and smooth function in the neighborhood $\mathcal{U} = \{X \in \mathbb{S}^m : \lambda_{\bar{r}}(X) > \lambda_{\bar{r}+1}(X)\}$ of $\mathcal{M}_{\bar{r}}$. Then we may replace the nonsmooth information contained in $f = \lambda_1 \circ \mathcal{F}$ by the smooth information contained in the function $\hat{f} = \hat{\lambda}_{\bar{r}} \circ \mathcal{F}$ by adding the constraint $\mathcal{F}(x) \in \mathcal{M}_{\bar{r}}$. We replace program (1) by the nonlinear *smooth* constrained program:

$$\begin{array}{ll} \text{minimize} & \hat{f}(x) = \hat{\lambda}_{\bar{r}}\left(\mathcal{F}(x)\right),\ x \in \mathbb{R}^n \\ \text{subject to} & \mathcal{F}(x) \in \mathcal{M}_{\bar{r}} \end{array} \tag{4}$$

A set of equations $h_1(X) = 0, \ldots, h_p(X) = 0$ describing the manifold $\mathcal{M}_{\bar{r}}$ has been presented independently in [4], [34], [46] and later in [38]. Here $p = \frac{\bar{r}(\bar{r}+1)}{2} - 1$ is the codimension of $\mathcal{M}_{\bar{r}}$ in $\mathbb{S}^m$. The set $\mathcal{N}_{\bar{r}} = \{x \in \mathbb{R}^n : \mathcal{F}(x) \in \mathcal{M}_{\bar{r}}\}$ is then described by the set of equations $h_1(\mathcal{F}(x)) = 0, \ldots, h_p(\mathcal{F}(x)) = 0$.

To apply standard SQP methods to (4), we require a constraint qualification hypothesis. Transversality used in [34,46,38,39] may serve this purpose. An easy way to describe it is to say that the Jacobian of the set of equations $h_1(\mathcal{F}(x)) = 0, \ldots, h_p(\mathcal{F}(x)) = 0$ has maximal rank p for x in a neighborhood of $\bar{x}$. In that event, $\mathcal{N}_{\bar{r}}$ is also a smooth manifold. Under transversality, local theory for (4) will therefore follow standard lines in SQP theory. All this has been presented in [32,34] and [46], and our only concern here is how the local second order theory goes along with the first-order part needed for global convergence.

Let us first examine the extra work required for the tangent quadratic program for (4). The Lagrangian is $L(x;\sigma) = \hat{\lambda}_{\bar{r}}(\mathcal{F}(x)) + \sum_{i=1}^p \sigma_i h_i(\mathcal{F}(x))$. Therefore, the tangent program used to compute the step δx about the current point x with current Lagrange multiplier estimate σ is

$$\begin{array}{ll} \text{minimize} & \hat{f}'(x)^\top \delta x + \frac{1}{2}\delta x^\top L''(x,\sigma)\,\delta x,\ \delta x \in \mathbb{R}^n \\ \text{subject to} & h_i(x) + h_i'(x)^\top \delta x = 0,\ i = 1, \ldots, p \end{array} \tag{5}$$

Let us find the explicit forms of the gradient $\hat{f}'(x)$ and the Hessian $L''(x,\sigma)$ of the Lagrangian L.

Following [34, Cor. 3.10], $\hat{f}$ is differentiable in the neighborhood $\mathcal{U}$ of $\mathcal{M}_{\bar{r}}$. An explicit formula for the gradient is

$$\hat{f}'(x) = \bar{r}^{-1}\, \mathcal{F}'(x)^* \left[\sum_{\ell \leq \bar{\ell}} Q_{k_\ell} Q_{k_\ell}^\top \right], \tag{6}$$

where k_ℓ are the indices of the leading eigenvalues, that is,

$$\lambda_{k_\ell - 1}(X) > \lambda_{k_\ell}(X) = \cdots = \lambda_{k_{\ell+1}-1}(X) > \lambda_{k_{\ell+1}}(X),$$

and where $k_{\bar\ell}$ is the index of the leading eigenvalue of the group containing $\lambda_{\bar r}$, which is itself at the end of this group, that is, $\bar r = k_{\bar\ell+1} - 1$. Here the columns of Q_{k_ℓ} form an orthonormal basis of the eigenspace of $\lambda_{k_\ell}(X)$.

The Hessian of the Lagrangian on the other hand is $L''(x,\sigma) = \hat f''(x) + \sum_{i=1}^p \sigma_i\, h_i''(x)$, where the second derivative of the objective function is characterized by its quadratic form

$$d^\top \hat f''(x) d = \hat\lambda_{\bar r}'(X) \bullet [d, \mathcal{F}''(x)d] + D \bullet \hat\lambda_{\bar r}''(X)\, D,$$

and the quadratic form of $\hat\lambda_{\bar r}''(X)$ is explicitly given in [47, Prop. 1.3]:

$$D \bullet \hat\lambda_{\bar r}''(X)\, D = \frac{2}{\bar r} \sum_{\ell \le \bar\ell} \operatorname{tr}\left(Q_{k_\ell}^\top D \left(\lambda_{k_\ell} I_m - X\right)^\dagger D Q_{k_\ell} \right). \tag{7}$$

Finally, an explicit form of $h''(x)$ is presented in [46,34], where the approach (4) has been discussed among other eigenvalue programs.

The extra work involved in computing $\hat f'(x_k)$ is minor, as the matrix Q_{ϵ_k} needed for the steepest ϵ_k-enlarged descent direction in the first-order method requires the same work. Suppose we compute the Cauchy point x_k^C first, which involves computing the $r(\epsilon_k) \times m$ matrix Q_{ϵ_k}. Then we may re-arrange these columns into the different matrices Q_{ℓ_k}, if necessary by adding some columns. The extra work for the second-order method is therefore solely in computing the Hessian of the Lagrangian $L''(x;\sigma)$, and of course in solving the tangent quadratic program.

Remark. As a result of this section we have a first realization of the general scheme in Figure 1 in the case of program (4). The Cauchy point x^C may be provided by the method in [31] or any of the variants from [40,2,1,3]. The Newton step x^N is computed by solving the tangent quadratic program (5), combined either with a line-search [6], trust region [9] or filter method [15]. In each case, the progress of the model has to be compared to the progress of the function $\hat f$, which assures that x^N is found by a finite procedure. If no progress in $\hat f$ is possible, we fall back to x^C, and only if x^C provides no progress over the current x, we stop at a critical point of f.

6. Projected Newton method for $F(x) = 0$

In this section we consider a more elaborate form of the tangent program, which exploits the information obtained from the oracle a little further.

Provided we can trust our guess r of the limiting multiplicity $\bar r$ of λ_1, it seems attractive to force the iterates X_k to lie on the manifold $\mathcal{M}_r$, because this is where we expect $\bar X$ to lie in, and because we will then benefit from the

Projected Newton method for $F = 0$

1.	Given the current iterate x_c, compute a Newton step $x_+ = x_c - (F'(x_c))^{-1} F(x_c)$.
2.	Project x_+ orthogonally onto $\mathcal{M}$ and obtain $x_{++} \in \mathcal{M}$.
3.	Replace x_c by x_{++} and loop on with step 1.

Fig. 2.

differentiable structure of $\mathcal{M}_r$. In the following we examine this option from a slightly more general point of view.

Suppose we want to solve the nonlinear system of equations $F(x) = 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$. Suppose further that a little bird tells us that a local solution $\bar{x}$ lies on a manifold $\mathcal{M}$ described by the system of equations $G(x) = 0$, where $G : \mathbb{R}^n \to \mathbb{R}^k$ for some $k < n$. Then we may consider the projected Newton method proposed in Figure 2.

Bringing in the additional information $\bar{x} \in \mathcal{M}$ as in step 2 of the method seems attractive, but we have to make sure that this is not in conflict with Newton's method. We have the following

Proposition 1. *Suppose $F(\bar{x}) = 0$ and $G(\bar{x}) = 0$. Suppose $F'(\bar{x})$ is invertible with $\| (F'(\bar{x}))^{-1} \| \leq \beta$. Suppose further that $F'(x)$ is Lipschitz continuous with constant γ on a neighborhood $B(\bar{x}, r) = \{x : \|x - \bar{x}\| \leq r\}$ of $\bar{x}$. Then there exists $0 < \rho \leq r$ such that for every $x_1 \in B(\bar{x}, \rho)$, the sequence x_k generated by the projected Newton method in Figure 2 is well-defined, stays in $B(\bar{x}, \rho)$, and converges quadratically to $\bar{x}$.*

Proof. We closely follow [11, p. 90]. Let $\rho = \min\{r, 1/3\beta\gamma\}$. Then the argument in that reference shows that the Newton step x_+ at the current x_c satisfies

$$\|x_+ - \bar{x}\| \leq \frac{3}{4}\beta\gamma\|x_c - \bar{x}\|^2. \tag{8}$$

Since by induction $x_c \in B(\bar{x}, \rho)$, we have $\|x_c - \bar{x}\| \leq 1/3\beta\gamma$, hence

$$\|x_+ - \bar{x}\| \leq \frac{1}{4}\|x_c - \bar{x}\|.$$

By the definition of x_{++} in step 3 of the algorithm we have $\|x_+ - \bar{x}\| \geq \|x_+ - x_{++}\|$. Hence $\|x_{++} - \bar{x}\| \leq \|x_+ - \bar{x}\| + \|x_+ - x_{++}\| \leq 2\|x_+ - \bar{x}\|$, so we obtain

$$\|x_{++} - \bar{x}\| \leq \frac{1}{2}\|x_c - \bar{x}\|.$$

This last estimate proves $\|x_{++} - \bar{x}\| \leq \frac{1}{2}\rho$, so that iterates stay in the ball $B(\bar{x}, \rho)$. This proves that the projected Newton method in Figure 2 is well-defined, and that the sequence of iterates x_{++} converges linearly to $\bar{x}$. Going back to (8) with this information proves quadratic convergence. Indeed, we now have

$$\|x_{++} - \bar{x}\| \leq 2\|x_+ - \bar{x}\| \leq \frac{3}{2}\beta\gamma\|x_c - \bar{x}\|^2,$$

Local second-order algorithm for $f = \lambda_1 \circ \mathcal{F}$

1. Given the current iterates x, X and multiplier estimates σ, Σ, choose ϵ and $r = r(\epsilon)$. Project X onto $\mathcal{M}_r$ and obtain X_r.
2. Do a Newton step $(x, X_r, \sigma, \Sigma) + (\delta x, \delta X, \delta\sigma, \delta\Sigma)$ for the KKT system (10).
3. Obtain $(x_+, X_+, \sigma_+, \Sigma_+)$ and loop on with step 1.

Fig. 3.

which proves the claim.

We use the projected Newton method to approach the second-order program (4) from a different point of view. Consider the following equivalent cast of (4), lifted in the space $\mathbb{R}^n \times \mathbb{S}^m$:

$$\begin{aligned} &\text{minimize} \;\; \hat{\lambda}_{\bar{r}}(X) \\ &\text{subject to } h(X) = 0 \text{ (i.e. } X \in \mathcal{M}_{\bar{r}}) \\ &\qquad\qquad\;\; \mathcal{F}(x) - X = 0 \end{aligned} \tag{9}$$

The Lagrangian is $L(x, X; \sigma, \Sigma) = \hat{\lambda}_{\bar{r}}(X) + \sigma^\top h(X) + \Sigma \bullet (\mathcal{F}(x) - X)$, and the necessary optimality conditions read:

$$\hat{\lambda}'_{\bar{r}}(X) + h'(X)^*\sigma - \Sigma = 0, \quad \mathcal{F}'(x)^*\Sigma = 0, \quad h(X) = 0, \quad \mathcal{F}(x) - X = 0, \tag{10}$$

which defines a system of equations in $(x, X, \sigma, \Sigma) \in \mathbb{R}^n \times \mathbb{S}^m \times \mathbb{R}^p \times \mathbb{S}^m$ with a corresponding self-map F. But the oracle told us that $\bar{X} \in \mathcal{M}_{\bar{r}}$, the information we already used to come up with the second-order cast. So why not use it again by defining the manifold

$$\mathcal{M} = \mathbb{R}^n \times \mathcal{M}_{\bar{r}} \times \mathbb{R}^p \times \mathbb{S}^m,$$

described by the set of equations $G(x, X, \sigma, \Sigma) = h(X) = 0$, and stabilize or even accelerate Newton's method by projecting onto $\mathcal{M}$ as above? The projection operator onto $\mathcal{M}$ is readily found, it is $(x, X, \sigma, \Sigma) \mapsto (x, X_{\bar{r}}, \sigma, \Sigma)$, where $X \mapsto X_{\bar{r}}$ is the orthogonal projection onto $\mathcal{M}_{\bar{r}}$, which is given by (cf. [39, Section 5.3]):

$$X \mapsto X_{\bar{r}} := X + Q_\epsilon \left(\hat{\lambda}_{\bar{r}}(X) I_{\bar{r}} - \operatorname{diag}(\lambda_1(X), \ldots, \lambda_{\bar{r}}(X)) \right) Q_\epsilon^\top.$$

Here $r(\epsilon) = \bar{r}$, and the columns of Q_ϵ are an orthonormal basis of the invariant subspace of X associated with the first $r(\epsilon)$ eigenvalues of X. The corresponding projected Newton method is now the one given in Figure 3.

In [39] a similar approach is considered. Projecting $X \to X_r$ is referred to as a *vertical step*, while the Newton type iteration is called a *tangential step*. Deriving these steps from the general projected Newton method above not only proves local convergence of the method, but clarifies the outset.

Naturally we should cast the Newton step as a tangent quadratic program. Its appealing feature is that due to linearity of the constraint $\mathcal{F}(x) - X = 0$ in X, we may eliminate the variable δX and obtain a program in δx. Omitting constant terms, this becomes

$$\begin{array}{ll} \text{minimize} & \left[\mathcal{F}'(x)^* \left(\hat{\lambda}_r'(X_r) + (\hat{\lambda}_r''(X_r) + h''(X_r)\sigma)(\mathcal{F}(x) - X_r)\right)\right]^\top \delta x \\ & +\frac{1}{2}\delta x^\top [\mathcal{F}'(x)^*(\hat{\lambda}_r''(X_r) + h''(X_r)\sigma)\mathcal{F}'(x) + \mathcal{F}''(x)\Sigma]\,\delta x \\ \text{subject to} & h(X_r) + h'(X_r)^*(\mathcal{F}(x) - X_r) + \mathcal{F}'(x)\delta x = 0 \end{array} \tag{11}$$

After obtaining the step δx and the multiplier update $\sigma + \delta\sigma$ as the multiplier of the constraint in (11), we obviously obtain the step $\delta X = \mathcal{F}'(x)\delta x$ in matrix space, while the matrix multiplier update $\Sigma + \delta\Sigma$ is obtained as

$$\Sigma^+ = \Sigma + \delta\Sigma = h'(X^+)^\star(\sigma + \delta\sigma) + \sigma^\top h''(X^+)\delta X,$$

a relation which is explicit since X is explicit in the artificial constraint $\mathcal{F}(x) - X = 0$. Program (11) is interesting since it contains an element which we already encountered in the first part [31].

Lemma 1. *Suppose $r = r(\epsilon) = \bar{r}$. Then for X sufficiently close to $\bar{X}$, $\hat{\lambda}_r'(X_r)$ is just the smallest ϵ-enlarged subgradient of λ_1 at X, i.e., $\hat{\lambda}_r'(X_r) = \operatorname{argmin}\{\|G\| : G \in \delta_\epsilon\lambda_1(X)\}$.*

Proof. Since $r(\epsilon) = r = \bar{r}$, [39, Prop. 9] shows that we have $\delta_\epsilon\lambda_1(X) = \partial\lambda_1(X_r) = \{Q_\epsilon Y Q_\epsilon^\top : Y \succeq 0, \operatorname{tr}(Y) = 1, Y \in \mathbb{S}^r\}$, where the latter uses the well-known characterization of $\partial\lambda_1$. Here Q_ϵ is of size $r \times m$ and its columns form an orthonormal basis of the eigenspace of $\lambda_1(X_r)$. Therefore the program defining steepest ϵ-enlarged descent is

$$\min\{\|Q_\epsilon Y Q_\epsilon^\top\| : Y \succeq 0,\ \operatorname{tr}(Y) = 1,\ Y \in \mathbb{S}^r\}.$$

Observe next that the function $\hat{\lambda}_r$ is differentiable in a neighborhood of $\bar{X}$, hence at X_r in this neighborhood. As we have $\lambda_1(\bar{X}) = \cdots = \lambda_r(\bar{X}) > \lambda_{r+1}(\bar{X})$ at the optimum $\bar{X}$, we also have $\lambda_r(X) > \lambda_{r+1}(X)$ as soon as X is sufficiently close to $\bar{X}$. In particular, X_r, the projection of X onto $\mathcal{M}_r$, is then also close to $\bar{X}$, and we deduce that $\lambda_1(X_r) = \cdots = \lambda_r(X_r) > \lambda_{r+1}(X_r)$. Then formula (6) simplifies to

$$\hat{\lambda}_r'(X_r) = r^{-1}Q_\epsilon Q_\epsilon^\top,$$

because the only relevant eigenvalue gap of X_r is between r and $r+1$, and since $r = r(\epsilon)$. It suffices now to argue that $\|Q_\epsilon Y Q_\epsilon^\top\| = \|Y\|$ and that the minimum over these $\|Y\|$ with $Y \succeq 0$ and $\operatorname{tr}(Y) = 1$ is attained at $Y = I_r$. In fact, this comes down to

$$\min\left\{\sum_{i=1}^r t_i^2 : \sum_{i=1}^r t_i = 1,\ t_i \geq 0\right\},$$

which is attained at $t_1 = \cdots = t_r = 1/r$.

Computing $\hat{\lambda}_r'(X_r)$ is even closer to what is required for the first-order method. The matrix Q_ϵ is directly used for both, which is a slight advantage of the projection method over the approach (4). Similarly, $\hat{\lambda}_r''(X_r)$ in (7) comes out somewhat simpler. Since the artificial variable δX may be eliminated in the tangent program, this means that the additional projection step does not increase the numerical burden of the method.

REMARK. Notice that $g_r = \mathcal{F}'(x)^* \hat{\lambda}_r'(X_r)$ is the image of the minimum norm element $G_r = \hat{\lambda}_r'(X_r)$ in $\delta_\epsilon \lambda_1(X)$ under $\mathcal{F}'(x)^*$, while the steepest ϵ-enlarged subgradient is the minimum norm element amongst the $g = \mathcal{F}'(x)^* G$, $G \in \delta_\epsilon \lambda_1(X)$. This looks pretty close, as if only a change of norms was involved. So could we use g instead of g_r in the second-order method? This would be convenient, as g is required for the first-order algorithm. The answer is no, as can be seen from the fact that g will tend to 0, while g_r, the gradient of the objective in (4), will only converge to 0 when $\bar{x}$ is an unconstrained minimum, a case we exclude when we assume $\bar{r} > 1$.

We conclude this part by relating the local algorithm in Figure 3 to the general scheme from Figure 1. While x^C is obtained as before, we now use the tangent quadratic program (11) to compute the Newton step x^N. Progress of x^N over the current x has to be evaluated with regard to the guessed model $\hat{f}$ upon which (9) is built. When x^N does not offer sufficient progress over the current x, the second order method is dispensed with, and the first-order step x^C is taken. In this event, the second order model is restarted at the next sweep with a new guess r_+ based upon (3). Notice that when $r_+ \neq r$, new Lagrange multiplier estimates will be required at the next iteration.

7. The affine case

In this section we compare the projected Newton approach to previous work in [39] for affine $\mathcal{A}$. Here we have the choice to representing the tangent quadratic program to (4) *either* in the variable δx as above, *or* in the variable δX. In the latter it looks as follows:

$$\begin{array}{ll} \text{minimize} & \hat{\lambda}_r'(X_r) \bullet \delta X + \frac{1}{2}\delta X \bullet \left(\hat{\lambda}_r''(X_r) + h''(X_r)\sigma\right) \delta X \\ \text{subject to} & h(X_r) + h'(X_r)^* \delta X = 0 \\ & A_0 - X_r - \delta X \in \text{range}(A) \end{array} \tag{12}$$

First observe that the Hessian (7) has a more amenable form at points $X_r \in \mathcal{M}_r$:

$$D \bullet \hat{\lambda}''(X_r)\, D = 2r^{-1} \text{tr} \left(Q_\epsilon^\top D \left(\lambda_1(X) I_m - X_r\right)^\dagger D Q_\epsilon \right).$$

This is again due to the fact that the only relevant eigenvalue gap is between $k_\ell - 1 = r$ and $k_\ell = r + 1$. Now let us recall the following

Definition 1. *(cf. Oustry [37,38], [29]). Let $X \in \mathcal{M}_r$ and $G \in \partial\lambda_1(X)$ and define a linear operator $H(X,G) : \mathbb{S}^m \to \mathbb{S}^m$ by*

$$H(X,G)D = GD\left(\lambda_1(X)I_m - X\right)^\dagger + \left(\lambda_1(X)I_m - X\right)^\dagger DG.$$

Let $\mathcal{U}(X) = \{U \in \mathbb{S}^m : Q_1^\top U Q_1 - r^{-1}\left(\operatorname{tr} Q_1^\top U Q_1\right) I_r = 0\}$, where the columns of Q_1 form an orthonormal basis of the eigenspace of $\lambda_1(X) = \cdots = \lambda_r(X)$. Then

$$\nabla^2 L_{\mathcal{U}}(X,G;0) = \operatorname{proj}^*_{\mathcal{U}(x)} H(X,G) \operatorname{proj}_{\mathcal{U}(X)}$$

is called the $\mathcal{U}$-Hessian of λ_1 at (X,G), where $\operatorname{proj}_{\mathcal{U}(X)}$ is the orthogonal projection onto $\mathcal{U}(X)$.

In [38,39] the $\mathcal{U}$-Hessian is obtained from the more general $\mathcal{U}$-Lagrangian theory developed in [29]. We reproduce it here in a slightly less general form, which is sufficient for our purpose. Indeed, observe that in our case, $r(\epsilon) = r$ and $Q_\epsilon = Q_1$. Therefore, we have the following

Lemma 2. *Let $r(\epsilon) = r$. Let $X_r \in \mathcal{M}_r$ and let $G_r = \hat\lambda_r'(X_r) \in \delta_\epsilon\lambda_1(X) = \partial\lambda_1(X_r)$ be the steepest ϵ-enlarged subgradient. Then*

1. *The operator $H(X_r, G_r)$ is identical with the Hessian $\hat\lambda_r''(X)$.*
2. *$\mathcal{U}(X_r)$ is the tangent space to $\mathcal{M}_r$ at X_r.*
3. *For $D \in \mathcal{U}(X_r)$, the $\mathcal{U}$-Hessian and $\hat\lambda_r''(X)$ agree.*

Proof. By the definition of $H(X_r, G_r)$ we have

$$D \bullet H(X_r,G_r)D = 2\operatorname{tr}\left(G_r D\left(\lambda_1(X_r)I_m - X_r\right)^\dagger D\right).$$

Now recall that $G_r = \hat\lambda_r'(X_r) = r^{-1}Q_\epsilon Q_\epsilon^\top$, hence

$$D \bullet H(X_r,G_r)D = 2r^{-1}\operatorname{tr}\left(G_\epsilon^\top D\left(\lambda_1(X)I_m - X_r\right)^\dagger DQ_\epsilon\right) = D \bullet \hat\lambda_r''(X_r)D,$$

which proves equality in item 1. Items 2 and 3 are now clear.

This brings us now to the following observation. In [39, 5.4 (51)] the author proposes the following tangent program for the affine case $f = \lambda_1 \circ \mathcal{A}$, $\hat f = \hat\lambda_r \circ \mathcal{A}$.

$$(*)\quad \begin{array}{ll} \text{minimize} & G_r \bullet \delta X + \frac{1}{2}\delta X \bullet H(X_r,G_r)\,\delta X \\ \text{subject to} & \delta X \in \mathcal{U}(X_r) \\ & X_r + \delta X \in A_0 + \operatorname{range}(A) \end{array}$$

From Lemma 2, item 1, we now see that programs (12) and (*) are the same, up to omission of the term $h''(X_r)\sigma$ in $(*)$ above. As Proposition 2 in the next section will show, this omission entails that $(*)$ above and therefore Theorem 15 in [39] is incorrect. The same happens in program (3.47) of [28], where the omission of the constraint Hessian foils quadratic convergence of the second-order proximal bundle algorithm 3.3 presented in that reference. Yet another instance of this error is [18], where the tangent program (11.7.38) misses the very same term. The error may finally be traced back to Algorithm 6.12 in [38], and to Theorem 6.13 of that paper.

Projected quasi-Newton method for $F = 0$

1.	Given the iterate x_k, compute a quasi-Newton step $\hat{x}_{k+1} = x_k - A_k^{-1} F(x_k)$.
2.	Project $\hat{x}_{k+1}$ onto $\mathcal{M}$ and obtain x_{k+1}.
3.	Update $A_k \to A_{k+1}$, and loop on with step 1.

Fig. 4.

8. Projected quasi-Newton method for $F(x) = 0$

In order to examine which digressions from the model projected Newton method in Figure 2 are authorized without foiling superlinear convergence, we need a result in the spirit of the classical Dennis-Moré theorem on quasi-Newton methods. We introduce a projected quasi-Newton method via Figure 4.

By Proposition 1 this scheme exhibits local quadratic convergence in the default case $A_k = F'(x_k)$. We prove an extension to quasi-Newton methods under the hypothesis that the projection P onto the manifold $\mathcal{M}$ is a differentiable operator.

Proposition 2. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$, and let the solution $\bar{x}$ of $F(x) = 0$ be an element of the manifold $\mathcal{M}$. Suppose F' is Lipschitz continuous on a neighborhood D of $\bar{x}$, and that $F'(\bar{x})$ is nonsingular. Suppose the orthogonal projector P onto $\mathcal{M}$ is differentiable. Let A_k be a sequence of quasi-Newton matrices such that the sequences x_k, $\hat{x}_k$ generated by the projected quasi-Newton algorithm remain in D. Then x_k converges superlinearly to $\bar{x}$ if and only if*

$$\lim_{k\to\infty} \frac{\| (A_k - F'(\bar{x}))\, (x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \tag{13}$$

Proof. 1) Assume that the sequence x_k converges superlinearly to $\bar{x}$. Consider all possible choices of matrices R_k such that

$$x_{k+1} = P\left(x_k - A_k^{-1} F(x_k)\right) = x_k - R_k^{-1} F(x_k).$$

Equivalently, $R_k \left(x_k - P\left(x_k - A_k^{-1} F(x_k)\right)\right) = F(x_k)$, so R_k is not fixed by this secant equation. Therefore choose R_k so that it is closest in norm to a given matrix B. By the well-known Broyden formula this gives

$$R_k = B + \frac{(z_k - Bw_k) w_k^\top}{w_k^\top w_k}, \quad w_k = x_k - P\left(x_k - A_k^{-1} F(x_k)\right),\ z_k = F(x_k).$$

Notice that for almost every regular matrix B, the matrices R_k will be regular for some k_0 and all $k \geq k_0$.

With any of these R_k, the projected quasi-Newton method based on A_k has now become a standard quasi-Newton method without projection. We may therefore invoke the Dennis-Moré theorem [11, Thm. 8.2.4]. It gives

$$\lim_{k\to\infty} \frac{\| (R_k - F'(\bar{x}))\, (x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0.$$

In order to prove (13), it suffices to show that for every subsequence $k \in \mathcal{K}$ such that

$$\lim_{k\in\mathcal{K}} \frac{x_{k+1}-x_k}{\|x_{k+1}-x_k\|} = \bar{d} \text{ and } \lim_{k\in\mathcal{K}} A_k = \bar{A}$$

we must have $\bar{A}\bar{d} = F'(\bar{x})\bar{d}$. Moreover, passing to another subsequence $\mathcal{K}' \subset \mathcal{K}$, we may assume that $R_k \to \bar{R}$ $(k \in \mathcal{K}')$ for some $\bar{R}$. Then by the above we have $\bar{R}\bar{d} = F'(\bar{x})\bar{d}$. Passing to yet another subsequence $\mathcal{K}'' \subset \mathcal{K}'$ if necessary, we may also assume that

$$\lim_{k\in\mathcal{K}''} \frac{A_k^{-1}F(x_k)}{\|A_k^{-1}F(x_k)\|} = \hat{d}$$

for some unit vector $\hat{d}$. Now observe that

$$\frac{z_k}{\|x_k-\bar{x}\|} = \frac{F(x_k)-F(\bar{x})}{\|x_k-\bar{x}\|} \to F'(\bar{x})\bar{d},$$

Furthermore, since $x_k = P(x_k)$,

$$\frac{w_k}{\|x_k-\bar{x}\|} = -\frac{P(x_k - A_k^{-1}F(x_k)) - P(x_k)}{\|-A_k^{-1}F(x_k)\|} \cdot \frac{\|A_k^{-1}F(x_k)\|}{\|x_k-\bar{x}\|}.$$

We deduce that

$$\frac{w_k}{\|x_k-\bar{x}\|} \to \|\bar{A}^{-1}F'(\bar{x})\bar{d}\| \cdot P'(\bar{x})\hat{d} \qquad (k \in \mathcal{K}'')$$

Now using the definition of R_k we see that $\bar{R}\bar{d} = F'(\bar{x})\bar{d}$ implies

$$F'(\bar{x})\bar{d} = B\bar{d} + \frac{\left(F'(\bar{x})\bar{d} - \alpha BP'(\bar{x})\hat{d}\right)\alpha\left(P'(\bar{x})\hat{d}\right)^\top}{\alpha^2\|P'(\bar{x})\hat{d}\|^2},$$

with $\alpha := \|\bar{A}^{-1}F'(\bar{x})\bar{d}\|$. Setting $\beta := \|P'(\bar{x})\hat{d}\|$ and $\bar{v} := P'(\bar{x})\hat{d}$, $\gamma = \bar{v}^\top\bar{d}$, this identity becomes

$$F'(\bar{x})\bar{d} = B\left(\bar{d} - \frac{\gamma}{\beta}\bar{v}\right) + \frac{1}{\alpha^2\beta^2}F'(\bar{x})\bar{d}. \tag{14}$$

Now observe that the constants α, β, γ do not depend on the choice of the matrix B, and neither do $\bar{d}$, $\hat{d}$ and $\bar{v}$. Since the only request on B was that the R_k were invertible, we may choose B arbitrary in a dense set of matrices, with (14) still satisfied for the same data α, β, γ and $\bar{v}$, as those did not depend on B. We deduce that in (14), we must have $\alpha\beta = 1$ and $\bar{d} = \frac{\gamma}{\beta}\bar{v}$. The latter gives

$$\bar{d} = \gamma\frac{P'(\bar{x})\hat{d}}{\|P'(\bar{x})\hat{d}\|},$$

hence $\gamma = 1$. By the definition of γ that means $\bar{v}^\top\bar{d} = 1$. But notice that as a projection operator, P has Lipschitz constant 1, hence $\|P'(\bar{x})\| \le 1$, giving

$\|\bar{v}\| \leq \|P'(\bar{x})\|\|\hat{d}\| \leq 1$. But the only vector v with norm $\|v\| \leq 1$ having $v^\top \bar{d} = 1$ is $v = \bar{d}$. Hence $\bar{v} = \bar{d}$ or rather $P'(\bar{x})\hat{d} = \bar{d}$. This implies $\|P'(\bar{x})\hat{d}\| = \|\hat{d}\| = 1$.

Now observe that $P'(\bar{x})$ is the orthogonal projection onto the tangent space $T(\mathcal{M}, \bar{x})$ of $\mathcal{M}$ at $\bar{x}$. So vectors r having $\|P'(\bar{x})r\| = \|r\|$ must be parallel to the tangent space, and in that event, already $P'(\bar{x})r = r$. This implies $\hat{d} = \bar{d}$.

By the definition of $\hat{d}$ we have

$$\hat{d} = \lim_{k \in \mathcal{K}''} \frac{A_k^{-1}\left(F(x_k) - F(\bar{x})\right)}{\|x_k - \bar{x}\|} \cdot \frac{\|x_k - \bar{x}\|}{\|A_k^{-1}\left(F(x_k) - F(\bar{x})\right)\|} = \frac{\bar{A}^{-1}F'(\bar{x})\bar{d}}{\|\bar{A}^{-1}F'(\bar{x})\bar{d}\|}.$$

Since $\alpha = \|\bar{A}^{-1}F'(\bar{x})\bar{d}\| = 1$, we have $\hat{d} = \bar{A}^{-1}F'(\bar{x})\bar{d}$. But as we have seen, $\hat{d} = \bar{d}$, so $\bar{A}\bar{d} = F'(\bar{x})\bar{d}$ as claimed.

2) Conversely, starting with condition (13), we can prove that x_k converge superlinearly, by introducing matrices R_k as above, and showing that the Dennis-Moré condition will be satisfied for R_k. This amounts to reading the first part of the proof backwards. We skip over the details.

The result may seem puzzling at first, since the condition appears to be exactly the same as the classical Dennis-Moré condition, where no projection onto $\mathcal{M}$ occurs. The possibility of pushing the intermediate steps $\hat{x}_{k+1}$ toward the limit $\bar{x}$ by projecting $\hat{x}_{k+1} \mapsto x_{k+1}$ would seem to suggest that the choice A_k could be allowed more freedom. And indeed, condition (13) *is* less restrictive than the classical Dennis-Moré condition. In fact, consider the limiting condition $\bar{A}\bar{d} = F'(\bar{x})\bar{d}$, arising in the proof. Using (13) we can assure this to hold for all unit vectors $\bar{d}$ arising as limits of quotients $\frac{x_{k+1}-x_k}{\|x_{k+1}-x_k\|}$. Since iterates x_k lie in $\mathcal{M}$, these $\bar{d}$ are necessarily tangent vectors to $\mathcal{M}$ at $\bar{x}$. Condition (13) is therefore less binding than its classical alter ego, where we have in principle to guarantee $\bar{A}\bar{d} = F'(\bar{x})\bar{d}$ for *all* $\bar{d}$, because without the presence of the manifold $\mathcal{M}$, the limiting directions $\bar{d}$ could be arbitrary. The fact that $\bar{d} \in T(\mathcal{M}, \bar{x})$ does give the matrices A_k more freedom to move $\hat{x}_k$ in directions transversal to $\mathcal{M}$. In order to highlight this consider the following

Example. Let $\mathcal{M}$ be a linear subspace of $\mathbb{R}^n$, P the linear orthogonal projection on $\mathcal{M}$. Fix any linear operator Q such that $P \circ Q = 0$. Given any sequence A_k satisfying (13), we generate another sequence $\tilde{A}_k$ such that $\tilde{A}_k^{-1} = A_k^{-1} + \rho_k Q$. Then $x_k - \tilde{A}_k^{-1}F(x_k) = x_k - A_k^{-1}F(x_k) - \rho_k QF(x_k)$. Since P is linear, this implies $P\left(x_k - \tilde{A}_k^{-1}F(x_k)\right) = P\left(x_k - A_k^{-1}F(x_k)\right) - \rho_k PBF(x_k) = P\left(x_k - A_k^{-1}F(x_k)\right)$, meaning that the iterates x_k generated by $\tilde{A}_k$ and A_k are the same, while the $\hat{x}_k$ differ. Playing with the factors ρ_k, we may arrange that $\hat{x}_k$ generated by the $\tilde{A}_k$ converge to $\bar{x}$ with arbitrarily slow speed, or even, fail to converge. This shows that superlinear convergence of the x_k need not imply superlinear convergence of $\hat{x}_k$. The argument of Proposition 1 on the other hand shows that if $\hat{x}_k$ converge superlinearly, the same is true for the x_k.

9. The constrained program (2)

Let us use our findings to analyze the constraint eigenvalue program (2). A smooth version based on the eigenvalue multiplicity oracle is obtained by the same mechanism. We replace (2) by

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{subject to} & \hat\lambda_r(\mathcal{F}(x)) \leq 0 \\ & \mathcal{F}(x) \in \mathcal{M}_r \end{array} \tag{15}$$

which may be treated by standard SQP methods. As this follows the usual lines, let us examine the more interesting case where intermediate projections on a smooth manifold are used.

Let us look at the lifted version of (15), which allows to bring in the projection onto $\mathcal{M}_r$ in alternance with the tangent step. Writing the program artificially in the space $\mathbb{R}^n \times \mathbb{S}^m$ gives

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{subject to} & \hat\lambda_r(X) \leq 0 \\ & h(X) = 0 \text{ (i.e. } X \in \mathcal{M}_r) \\ & \mathcal{F}(x) - X = 0 \end{array} \tag{16}$$

with the associated Lagrangian $L(x, X; \tau, \sigma, \Sigma) = c^\top x + \tau\hat\lambda_r(X) + \sigma^\top h(X) + \Sigma \bullet (\mathcal{F}(x) - X)$. The KKT conditions are

$$c + \mathcal{F}'(x)^\star\Sigma = 0, \tau\hat\lambda_r'(X) + h'(X)^\star\sigma - \Sigma = 0,$$
$$h(X) = 0, \mathcal{F}(x) - X = 0, \tau \geq 0, \hat\lambda_r(X) \leq 0, \tau\hat\lambda_r(X) = 0$$

The usual tangent quadratic program may be written in δx, δX or in the joint variable $(\delta x, \delta X)$. In δx space we obtain

$$\begin{array}{ll} \text{minimize} & c^\top \delta x + \frac{1}{2}\delta x^\top \left[\mathcal{F}'(x)^\star h''(X)\sigma\mathcal{F}'(x) + \mathcal{F}''(x)\Sigma\right]\delta x \\ \text{subject to} & \hat\lambda_r(X) + \hat\lambda_r'(X)\mathcal{F}'(x)\delta x \leq 0 \\ & h(X) + h'(X)\mathcal{F}'(x)\delta x = 0 \end{array} \tag{17}$$

As usual, the new Lagrange multipliers $\tau^+ = \tau + \delta\tau \geq 0$, $\sigma^+ = \sigma + \delta\sigma$ are the multipliers of the corresponding constraints in (17). The update X^+ is obtained as $\mathcal{F}(x^+)$. Finally, due to the special structure of the artificially augmented program, the matrix multiplier update $\Sigma^+ = \Sigma + \delta\Sigma$ is obtained as

$$\Sigma^+ = h'(X^+)^\star\sigma^+ + \tau^+\hat\lambda_r'(X^+) + \left[\tau^+\hat\lambda_r''(X^+) + h''(X^+)^\star\sigma^+\right]\delta X. \tag{18}$$

Similar to (4) in Section 6 we may now use an intermediate projection step $X \mapsto X_r$ in order to keep iterates close to the manifold $\mathcal{M}_r$ on which we expect the limit $\bar X$ to lie. This leads to the scheme in Figure 5.

Convergence analysis of the projected Newton scheme follows known lines. We have the following result, based on Bonnans [5].

Local second-order algorithm for (2)

1. Given iterate x, $X = \mathcal{F}(x)$ and Lagrange multiplier estimates $\tau \geq 0$, σ, obtain an estimate r of the limiting multiplicity $\bar{r}$ of λ_1.
2. Compute the orthogonal projection X_r of X onto $\mathcal{M}_r$.
3. Compute the matrix multiplier estimate Σ via (18).
4. Solve tangent quadratic program (17) and obtain the Newton step $(x, \tau, \sigma) + (\delta x, \delta\tau, \delta\sigma)$, where $\tau + \delta\tau \geq 0$, $\sigma + \delta\sigma$ are the multipliers in (17).
5. Compute $x_{++} = x + \alpha(x_+ - x)$ via line search. Adjust τ_{++}, σ_{++} and compute $X_{++} = \mathcal{F}(x_{++})$.
6. Replace x by x_{++}, τ by τ_{++}, σ by σ_{++}, and loop on with step 1.

Fig. 5.

Theorem 1. *Let $\bar{x}$ be a local minimum of (2). Let $\bar{r}$ the multiplicity of $\lambda_1(\bar{X})$, so that $\bar{x}$ is a local minimum of (15) with $r = \bar{r}$. Suppose that in (16) the gradients of the active constraints at $\bar{x}$ are linearly independent. Let $\bar{\tau} \geq 0$, $\bar{\sigma}$, $\bar{\Sigma}$ be the Lagrange multipliers associated with $\bar{x}, \bar{X}$ and suppose the second-order sufficient optimality condition is satisfied. Let $(x_k, X_k, \tau_k, \sigma_k, \Sigma_k)$ be the sequence generated by the local second-order algorithm based on tangent program (17), where the estimated eigenvalue multiplicity is $r_k = \bar{r}$, and where an intermediate projection step $X \mapsto X_{\bar{r}} \in \mathcal{M}_{\bar{r}}$ is performed, so that each $X_k \in \mathcal{M}_{\bar{r}}$. Then the sequence $(x_k, X_k, \tau_k, \sigma_k, \Sigma_k)$ converges quadratically to $(\bar{x}, \bar{X}, \bar{\tau}, \bar{\sigma}, \bar{\Sigma})$.*

Proof. We follow the argument in [5], where the KKT-system is embedded in a variational inequality, and the tangent quadratic program is interpreted as a Newton step for that variational inequality. Adopting the notation of section 2 of that reference, where the variable z replaces $(x, X, \tau, \sigma, \Sigma)$, we first observe that the second-order sufficient optimality condition implies strong regularity in the sense of Robinson and therefore semi-stability and hemi-stability in the sense of [5]. It therefore remains to accommodate the projection step $X \mapsto X_r$ in the analysis of [5, Theorem 2.2].

Observe that the projection step $X \mapsto X_{\bar{r}}$ corresponds to an orthogonal projection step $z \mapsto z_{\bar{r}}$ onto the manifold $\mathcal{M} = \mathbb{R}^n \times \mathcal{M}_{\bar{r}} \times \mathbb{R} \times \mathbb{R}^p \times \mathbb{S}^m$. Starting with $\epsilon_0 \leq \min(c_1, 1/4c_2)$, instead of $1/3c_2$ in the proof of [5, Theorem 2.2], we obtain the estimate $\|z_+ - \bar{z}\| \leq \frac{1}{3}\|z - \bar{z}\|$ (instead of $\leq \frac{1}{2}\|z - \bar{z}\|$ there), where z_+ is the usual Newton step away from z. Now let z_{++} be the projection of z_+ onto $\mathcal{M}$, then obviously $\|z_{++} - \bar{z}\| \leq \|z_{++} - z_+\| + \|z_+ - \bar{z}\| \leq \|\bar{z} - z_+\| + \|z_+ - \bar{z}\| = 2\|z_+ - \bar{z}\|$, where the second inequality uses the fact that $\bar{z} \in \mathcal{M}$. So we obtain the estimate $\|z_{++} - \bar{z}\| \leq \frac{2}{3}\|z - \bar{z}\|$, (replacing $\leq \frac{1}{2}\|z - \bar{z}\|$ in [5]). Then $\|z_{++} - z_+\| \leq 3\epsilon$, which proves that the sequence stays in the ball of radius 3ϵ (instead of 2ϵ in [5]). The argument now remains the same as in [5] and shows that the Newton iterates are well-defined and converge with quadratic speed.

Combined algorithm for (2)

1. Given the current iterate x, compute a Cauchy point x^C using the first-order spectral bundle method from [31]. Compute progress $p^C \geq 0$ of x^C over x using the improvement function (19). If $p^C = 0$ stop.
2. Using $X^C = \mathcal{F}(x^C)$, compute estimate r of limiting multiplicity $\bar{r}$ using (3).
3. Compute orthogonal projection X_+ of X onto $\mathcal{M}_r$. Obtain new multiplier estimate Σ.
4. Solve tangent quadratic program (17) to compute Newton trial step $(\delta x, \delta X, \delta\tau, \delta\sigma, \delta\Sigma)$ away from $(x, X^+, \tau, \sigma, \Sigma)$. Use merit function, trust region strategy or filter for program (15) to find step (x^N, τ^N, σ^N).
5. Compute progress p^N of x^N over x using the improvement function (19). If $p^N \geq \theta p^C$, let $x^+ = x^N$, $\tau^+ = \tau^N$, $\sigma^+ = \sigma^N$. Otherwise let $x^+ = x^C$. In the latter case, compute new multiplier estimates $\tau^+ \geq 0, \sigma^+$.
6. Replace old elements by ++ elements and go back to step 1.

Fig. 6.

10. Combined Algorithm

The elements in sections 6 and 9 may now be assembled to a globally convergent method which attempts second order steps x^N with the help of the oracle, based on an underlying first order method, which produces Cauchy points x^C to give a global convergence certificate.

The improvement function used at the first-order level is

$$\kappa(x, x^+) = \max\left(c^\top (x - x^+), f(x^+)\right) \tag{19}$$

cf. [22,31] or [20,7], but other choices are possible. For instance, an alternative which works for the maximum of a finite number of smooth functions is given in [40, Algorithm 2.4.1] under the name of an optimality function. An extension of that idea to maximum eigenvalue functions and to the H_∞-norm is considered in [1,3]. Alternatives for infinite maxima are considered in chapter 3 of [40]. If we work with (19), $p^C = 0$ respectively $\kappa = 0$ implies that x is a F. John stationary point. In [31, Sect. 4, Prop. 1] reasonable conditions are given which imply that x is even a KKT-point. This justifies the stopping test in step 1 of Figure 6.

As we have pointed out before, if step 4 in Figure 6 uses a trust region strategy, finite termination of the trust region radius updating should be based on program (15), while the final acceptation of x^N is controlled through x^C. Similarly, if a filter algorithm is used, it has to be guaranteed that the filter accepts some iterate after a finite number of steps. This is only possible if the smooth model (16) is temporarily used. As new entries, either $\left(c^\top x^C, \max\left(0, \lambda_1\left(\mathcal{F}(x^C)\right)\right)\right)$ or $\left(c^\top x^N, \max\left(0, \lambda_1\left(\mathcal{F}(x^N)\right)\right)\right)$ or even both are added afterward to give global convergence.

An additional difficulty arises when x^C is the new iterate, as is the case when x^N does not provide satisfactory progress due to failure to estimate $\bar{r}$ correctly. Here the multiplier estimates obtained in the tangent step are meaningless, and a good idea how to generate new $\tau^+ \geq 0$, σ^+ and Σ^+ at the next sweep is required. The situation is of course similar to the classical one in cases when the Newton steps is poor. In such a situation, multiplier estimates from (17) are also of bad quality. Similar numerical recipes may therefore be employed (cf. [9]).

References

1. P. Apkarian, D. Noll, *Nonsmooth H_∞-synthesis*. Submitted.
2. P. Apkarian, D. Noll, *Controller design via nonsmooth multi-directional search*. SIAM Journal on Control and Optimization.
3. P. Apkarian, D. Noll, *Nonsmooth optimization for multidisk H_∞ synthesis*, submitted.
4. V. I. Arnold, *On matrices depending on parameters*, Russ. Math. Surveys, 26 (1971), pp. 29 – 43.
5. J. F. Bonnans, *Local analysis of Newton-type methods for variational inequalities and nonlinear programming*, Appl. Math. Opt., 29 (1994), pp. 161 – 186.
6. J.F. Bonnans, E. Panier, A.L. Tits, J.L. Zhou, *Avoiding the Maratos effect by means of a nonmonotone line search II: Inequality problems - feasible iterates*, SIAM J. Num. Anal., vol. 29, no. 4, 1992, pp. 1187 – 1202.
7. Bui Trong Lieu, P. Huard, *La méthode des centres dans un espace topologique*, Numerische Mathematik, vol. 8, 1966, pp. 56 – 67.
8. F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.
9. A.R. Conn, N.I.M. Gould, P.L. Toint, *Trust-Region Methods*, MPS/SIAM Series on Optimization, 2000.
10. J. Cullum, W.E. Donath, P. Wolfe, *The minimization of certain nondifferential sums of eigenvalues of symmetric matrices,* Math. Progr. Stud., vol. 3 (1975), pp. 35 – 55.
11. J.E. Dennis jun., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, 1983.
12. R. Fletcher, *Semidefinite constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493 – 513.
13. R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, Chichester, 2nd ed., 1987.
14. R. Fletcher, S. Leyffer, *A bundle filter method for nonsmooth nonlinear optimization.* University of Dundee, Report NA/195, 1999.
15. R. Fletcher, S. Leyffer, *Nonlinear programming without a penalty function*, Math. Programming, vol. 91, 2002, pp. 239 – 269.
16. A. Fuduli, M. Gaudioso, G. Giallombardo, *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control*, SIAM J. on Optim., vol. 14, no. 3, 2005, pp. 743 – 756.
17. W.L. Hare, A.S. Lewis, *Identifying active constraints via partial smoothness and prox-regularity.* Journal of Convex Ananlysis, vol. 11, 2004, pp. 251 – 266.
18. C. Helmberg, F. Oustry, *Bundle methods to minimize the maximum eigenvalue function,* In: L. Vandenberghe, R. Saigal, H. Wolkowitz, (eds.), Handbook of Semidefinite Programming.
19. J.-B. Hiriart-Urruty, C. Lemaréchal, *Convex Analysis and Minimization Algorithms, part II,* Springer Verlag, Berlin, 1993.
20. P. Huard, *Resolution of mathematical programming with nonlinear constraints by the method of centers*, Nonlinear programming, P. Abadie (ed.), North Holland, 1967, pp. 209 – 219.
21. T. Kato, *Perturbation Theory for Linear Operators*. Springer Verlag, 1984.
22. K.C. Kiwiel, *Methods of descent for nondifferentiable optimization*, Lect. Notes in Math. vol. 1133, Springer Verlag, Berlin, 1985.
23. K.C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable optimization,* Math. Programming 46, 1990, pp. 105 – 122. Math. Programming, 46 (1990), pp. 105 – 122.
24. C. Lemaréchal, *Extensions diverses des méthodes de gradient et applications*, Thèse d'Etat, Paris, 1980.
25. C. Lemaréchal, *Bundle methods in nonsmooth optimization.* In: Nonsmooth Optimization, Proc. IIASA Workshop 1977, C. Lemaréchal, R. Mifflin (eds.), Pergamon Press, 1978.

26. C. Lemaréchal, *Nondifferentiable Optimization*, chapter VII in: *Handbooks in Operations Research and Management Science,* vol. 1, Optimization, G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd (eds.), North Holland, 1989.
27. C. Lemaréchal, A. Nemirovskii, Y. Nesterov, *New variants of bundle methods,* Math. Programming, 69 (1995), pp. 111 – 147.
28. C. Lemaréchal, F. Oustry,*Nonsmooth algorithms to solve semidefinite programs.* In: L. El Ghaoui, L. Niculescu (eds.), Advances in LMI Methods in Control, SIAM Series: Advances in Design and Control, 2000.
29. C. Lemaréchal, F. Oustry, C. Sagastizábal, *The $\mathcal{U}$-Lagrangian of a convex function*, Trans. Amer. Math. Soc. 352, 2000.
30. R. Mifflin, C. Sagastizábal, *VU-smoothness and proximal point results for some nonconvex functions.* Optimization Methods and Software, 2004, to appear.
31. D. Noll, P. Apkarian, *Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods.* Mathematical Programming Series B.
32. M.L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix,* SIAM J. Matrix Anal. Appl., vol. 9, no. 2 (1988), pp. 256 – 268.
33. M.L. Overton, *Large-scale optimization of eigenvalues,* SIAM J. Optim., vol. 2, no. 1 (1992), pp. 88 – 120.
34. M.L. Overton, R.S. Womersley,*Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices,* Math. Programming, 62 (1993), pp. 321 – 357.
35. M.L. Overton, R.S. Womersley, *Second derivatives for optimizing eigenvalues of symmetric matrices,* SIAM J. Matrix Anal. Appl., vol. 16, no. 3 (1995), pp. 697 – 718.
36. J.-P. A. Haeberly, M.L. Overton,*A hybrid algorithm for optimizing eigenvalues of symmetric definite pencils.* SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1141 – 1156.
37. F. Oustry, *Vertical development of a convex function*, J. Convex Analysis, 5 (1998), pp. 153 – 170.
38. F. Oustry, *The $\mathcal{U}$-Lagrangian of the maximum eigenvalue function,* SIAM J. Optim., vol. 9, no. 2 (1999), pp. 526 – 549.
39. F. Oustry, *A second-order bundle method to minimize the maximum eigenvalue function,* Math. Programming Series A, vol. 89, no. 1 (2000), pp. 1 – 33.
40. E. Polak, *Optimization. Algorithms and Consistent Approximations.* Springer Series in Applied Mathematical Sciences, vol. 124, Springer Verlag, 1997.
41. E. Polak, Y. Wardi, *Nondifferentiable optimization algorithm for designing control systems having singular value inequalities*, Automatica, vol. 18, no. 3, 1982, pp. 267 - 283.
42. C. Scherer, *A full block S-procedure with applications*, Proc. IEEE Conf. on Decision and Control, San Diego, USA, 1997, pp. 2602 – 2607.
43. C. Scherer, *Robust mixed control and linear parameter-varying control with full block scalings*, SIAM Series: Advances in Linear Matrix Inequality Methods in Control, L. El Ghaoui, S.I. Niculescu (eds.), 2000.
44. A. Shapiro, *Extremal problems on the set of nonnegative matrices*, Lin. Algebra and its Appl., 67 (1985), pp. 7 – 18.
45. A. Shapiro, *First and second-order analysis of nonlinear semidefinite programs*, Math. Programming Series B, 77 (1997), pp. 301 – 320.
46. A. Shapiro, M.K.H. Fan,*On eigenvalue optimization,* SIAM J. Optim., vol. 5, no. 3 (1995), pp. 552 – 569.
47. M. Torki,*First- and second-order epi-differentiability in eigenvalue optimization,* J. Math. Anal. Appl., 234 (1999), pp. 391 – 416.
48. Ph. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable convex functions*, Math. Programming Studies, 3 (1975), pp. 145 – 173.