Mathematical Programming 77 (1997) 163–190

The Projective Method for solving linear matrix inequalities

Pascal Gahinet [a,1], Arkadi Nemirovski [b,*]

[a] *INRIA Rocquencourt, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France*

[b] *Faculty of Industrial Engineering and Management, Technion, 32000 Technion City, Haifa, Israel*

Received 30 December 1994

Abstract

Numerous problems in control and systems theory can be formulated in terms of linear matrix inequalities (LMI). Since solving an LMI amounts to a convex optimization problem, such formulations are known to be numerically tractable. However, the interest in LMI-based design techniques has really surged with the introduction of efficient interior-point methods for solving LMIs with a polynomial-time complexity. This paper describes one particular method called the Projective Method. Simple geometrical arguments are used to clarify the strategy and convergence mechanism of the Projective algorithm. A complexity analysis is provided, and applications to two generic LMI problems (feasibility and linear objective minimization) are discussed. © 1997 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Linear matrix inequalities; Semidefinite programming; Interior point methods

1. Introduction

Linear matrix inequality (LMI) techniques are inspiring a growing interest in the control community and are emerging as powerful numerical tools for the analysis and design of control systems [4]. An LMI is any matrix inequality of the form

$$A(x) > 0$$

where $A(x)$ is a symmetric matrix that depends affinely on the entries of the vector $x \in \mathbb{R}^n$. In other words, $A(x)$ can be written as

$$A(x) = x_1 A_1 + \cdots + x_n A_n + B,$$

* Corresponding author. E-mail: nemirovs@ie.technion.ac.il

[1] E-mail: gahinet@colorado.inria.fr

0025-5610/97/$17.00
Published by Elsevier Science B.V.
PII S0025-5610(96)00085-8

where

- $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}$ are given symmetric matrices,
- $x = (x_1, \ldots, x_n)$ is a vector of real scalar variables,
- > 0 stands for positive definite.

The entries of x are often called the *decision variables*. Note that the system of LMIs

$$\begin{cases} L_1(x) > 0 \\ \quad \vdots \\ L_P(x) > 0 \end{cases}$$

is equivalent to the single LMI

$$L(x) := \mathrm{Diag}(L_1(x), \ldots, L_P(x)) > 0.$$

Hence the discussion below readily extends to systems of LMIs.

Two of the three generic LMI problems are discussed in this paper:

- The *strict feasibility problem*:

$$\text{Find } x \in \mathbb{R}^n \quad \text{such that } A(x) > 0. \tag{1}$$

  In other words, find values of the decision variables $x_1, \ldots, x_n$ that satisfy the LMI constraint $A(x) > 0$.
- The *linear objective minimization under LMI constraints*:

$$\text{Minimize } c^{\mathrm{T}}x \quad \text{subject to } A(x) \geqslant 0. \tag{2}$$

This problem only makes sense when the LMI constraint $A(x) \geqslant 0$ is feasible.

Both problems are convex due to the affine dependence in x. The third generic problem is the generalized eigenvalue minimization problem discussed in [3, 16].

Optimization under LMI constraints is a very attractive field for the application of modern polynomial-time interior-point methods. These methods are rooted in the seminal paper of Karmarkar [12] where the first method of this type was proposed for linear programming. Various extensions to semi-definite programming and optimization under LMI constraints can be found in [1, 3, 6, 11, 13–16, 19] and references therein. See also [17, 18] for alternative approaches to this class of problems.

This paper focuses on the so-called Projective Method. This interior-point method has several interesting features:

- As a polynomial-time algorithm, it is guaranteed to find, for any $\varepsilon > 0$, an ε-solution to the problem within a finite number of steps bounded by $O(m) \log(C/\varepsilon)$, where m is the total row size of the LMIs and C is some scaling factor (see Theorems 7 and 13 for details).
- The Projective Method is applicable to a variety of LMI problems including linear programs under LMI constraints and fractional problems of the form

$$\min \lambda \quad \text{subject to } A(x) \leqslant \lambda B(x), \quad B(x) > 0, \quad C(x) \geqslant 0.$$

- No initial feasible solution is required for problem (2), and no separate Phase-I algorithm needs to be run to generate such a solution.

The Projective Method was first described in [15, 16] in the more abstract context of the theory of self-concordant barriers. The present paper takes a different and more straightforward perspective and provides simple geometrical insight into the algorithm. Besides its tutorial value, the paper also contains details on the numerical implementation and statistics on the computational overhead. Note that a version of the Projective Method optimized for LMI problems with a block-matrix structure (see Section 5 is implemented in the *LMI Control Toolbox* for use with MATLAB [8,9].

The paper is organized as follows. Section 2 summarizes the notation and reviews a few instrumental concepts. Section 3 describes the Projective Algorithm for the strict feasibility problem. This includes a detailed proof of convergence as well as a complexity analysis. Section 4 shows how this basic algorithm can be adapted to solve the linear objective minimization problem. Finally, Section 5 discusses the implementation aspects with an emphasis on efficiency and numerical stability. The results of numerical tests are also reported.

2. Notation and preliminaries

The following notation is used in the sequel.

- $\mathrm{Tr}(X)$ and $\mathrm{Det}(X)$ denote the trace and the determinant of the square matrix X, and $\log(x)$ stands for the natural logarithm of the positive scalar x.
- $\mathcal{S}$ is the space of symmetric matrices of size m. This is a subspace of $\mathbb{R}^{m\times m}$ of dimension

$$M := \dim(\mathcal{S}) = \tfrac{1}{2}m(m+1). \tag{3}$$

- $\mathcal{K}$ denotes the open cone of positive definite matrices in $\mathcal{S}$, and $\bar{\mathcal{K}}$ denotes the closure of $\mathcal{K}$, that is, the cone of positive semi-definite matrices of $\mathcal{S}$.
- $\mathcal{S}$ is endowed with the family of scalar products $\langle .,. \rangle_P$ defined for $P > 0$ by

$$\langle X, Y \rangle_P = \mathrm{Tr}(PXPY).$$

The corresponding induced norms are

$$\|X\|_P = (\mathrm{Tr}(PXPX))^{1/2}.$$

The case $P = I$ yields the usual Frobenius norm which we denote by $\langle .,. \rangle_{\mathrm{Fro}}$. The origin and interpretation of the metrics $\langle .,. \rangle_P$ are clarified in the next subsection.

2.1. Logarithmic barriers

As an interior-point method, the Projective Method generates a sequence of matrices that remain in the open cone $\mathcal{K}$. To preserve positive definiteness, the optimization

criterion includes a logarithmic barrier that is defined on $\mathcal{K}$ and tends to $+\infty$ when approaching a boundary point of $\mathcal{K}$. This barrier is defined for $X > 0$ by

$$f(X) = -\log \mathrm{Det}(X) \tag{4}$$

and constitutes a special case of the "self-concordant" barriers discussed in [16]. It is easily verified that

$$f'(X) = -X^{-1}, \qquad [f''(X)](Y) = X^{-1}YX^{-1}$$

(here we identify linear and bilinear forms with their matrix representation). Consequently, the Hessian metric is given by

$$\langle [f''(X)](Y), Z\rangle_{\mathrm{Fro}} = \mathrm{Tr}(X^{-1}YX^{-1}Z) = \langle Y, Z\rangle_{X^{-1}}.$$

This explains the relevance of the inner products $\langle .,. \rangle_P$ introduced above.

2.2. *Dikin ellipsoid*

Instrumental to the updating of the current "best" solution is the notion of Dikin ellipsoid (see Fig. 1). Given $X > 0$, consider the ellipsoid centered at X:

$$\begin{aligned}\Omega(X) &= \left\{Y \mid \|Y - X\|^2_{X^{-1}} = \mathrm{Tr}\{X^{-1}(Y - X)X^{-1}(Y - X)\} < 1\right\} \\ &= \left\{Y \mid \|X^{-1/2}YX^{-1/2} - I\|_{\mathrm{Fro}} < 1\right\}.\end{aligned} \tag{5}$$

Lemma 1. *The Dikin ellipsoid $\Omega(X)$ is contained in $\mathcal{K}$.*

Proof. Let $Z := X^{-1/2}YX^{-1/2}$ and denote by $\{\lambda_i\}_{1\leqslant i\leqslant m}$ the eigenvalues of Z. It suffices to show that Z is positive definite whenever $\|Z - I\|_{\mathrm{Fro}} < 1$. Now

$$\|Z - I\|^2_{\mathrm{Fro}} = \sum_1^m (\lambda_i - 1)^2 < 1$$

implies that $|\lambda_i - 1| < 1$ for all i. Consequently, $\lambda_i(Z) > 0$ holds for all i if $\|Z - I\|_{\mathrm{Fro}} < 1$ and the proof is complete. □

The importance of the Dikin ellipsoid comes from the fact that it delimits a region around X where we can move without leaving the cone $\mathcal{K}$, i.e., losing positive

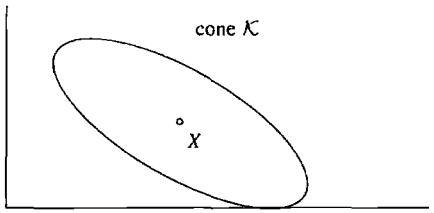

Fig. 1. Dikin ellipsoid.

definiteness. Note that the Dikin ellipsoid is exactly the open unit ball in the metric $\|.\|_{X^{-1}}$.

2.3. Orthogonal projection in the metric $\langle .,. \rangle_P$

An important tool in the Projective Method is the orthogonal projection onto a subspace E with respect to the metric $\langle .,. \rangle_P$. In our case, E will be the range of the linear map $\mathcal{A}$ associated with the LMI $A(x) > 0$ (see Section 3).

Given $X \in \mathcal{S}$, its projection X^+ onto E is defined as the unique solution of the least-squares problem

$$\min_{Y \in E} \|Y - X\|_P = \min_{x \in \mathbb{R}^n} \|\mathcal{A}x - X\|_P. \tag{6}$$

The gradient $g(Y)$ of the function $Y \mapsto \|Y - X\|_P^2$ is easily obtained from the first-order variation

$$\langle g(Y), \delta Y \rangle_{\text{Fro}} = \delta(\|Y - X\|_P^2) = 2\,\text{Tr}(P(Y - X)P\delta Y)$$

which yields

$$g(Y) = 2P(Y - X)P. \tag{7}$$

The projection X^+ is characterized by the optimality condition

$$\forall Y \in E, \quad \langle g(X^+), Y \rangle_{\text{Fro}} = 2\langle X^+ - X, Y \rangle_P = 0. \tag{8}$$

This is nothing else than saying that $X^+ - X$ should be orthogonal to E *with respect to the inner product* $\langle .,. \rangle_P$, or equivalently, that $P(X^+ - X)P$ should be orthogonal to E in the usual Frobenius metric $\langle .,. \rangle_{\text{Fro}}$. In the case $E = \text{Range}(\mathcal{A})$, the orthogonality condition (8) can be rewritten in terms of x^+ such that $X^+ = \mathcal{A}x^+$ as:

$$\forall x \in \mathbb{R}^n, \quad \langle \mathcal{A}x^+ - X, \mathcal{A}x \rangle_P = \langle P(\mathcal{A}x^+ - X)P, \mathcal{A}x \rangle_{\text{Fro}} = 0. \tag{9}$$

3. The strict feasibility problem

We first consider the strict feasibility problem (1). A vector x is called *strictly feasible* if $A(x) > 0$. The affine function $A(x)$ can be decomposed as

$$A(x) = \mathcal{A}x + \mathcal{B} = x_1\mathcal{A}_1 + \cdots + x_n\mathcal{A}_n + \mathcal{B} \tag{10}$$

where the $\mathcal{A}_i$'s and $\mathcal{B}$ are given matrices in $\mathcal{S}$, and $\mathcal{A}x$ is the homogeneous part of $A(x)$. Without loss of generality, we can assume $\mathcal{B} = 0$ for simplicity. To see this, introduce an extra variable τ and define

$$\tilde{\mathcal{A}} \begin{pmatrix} x \\ \tau \end{pmatrix} := \begin{pmatrix} \mathcal{A}x + \tau\mathcal{B} & 0 \\ 0 & \tau \end{pmatrix}.$$

Clearly, if x solves $\mathcal{A}x + \mathcal{B} > 0$, then $\tilde{x} := \binom{x}{1}$ solves $\tilde{\mathcal{A}}\tilde{x} > 0$. Conversely, given a solution $\binom{x}{\tau}$ of $\tilde{\mathcal{A}}\tilde{x} > 0$, τ is positive and x/τ satisfies $\mathcal{A}x + \mathcal{B} > 0$. Hence the two strict feasibility problems are equivalent.

Up to redefining $\mathcal{A}$, x, and n as $\tilde{\mathcal{A}}, \tilde{x}$, and $n+1$, respectively, we can therefore concentrate on the homogeneous strict feasibility problem:

$$\text{Find } x \in \mathbb{R}^n \quad \text{such that } \mathcal{A}x > 0, \tag{11}$$

where $\mathcal{A}$ is a linear map from $\mathbb{R}^n$ to $\mathcal{S}$. Henceforth, the range of this map is denoted by

$$E := \text{Range}(\mathcal{A}). \tag{12}$$

The geometrical interpretation of this problem is as follows:

> Given an open cone $\mathcal{K}$ with its vertex at the origin, and a plane E passing through the origin, find a matrix X in the intersection of E and $\mathcal{K}$.

Throughout the sequel, we require that

Nondegeneracy assumption. $\mathcal{A}x$ is nonzero whenever x is nonzero.

In other words, $\mathcal{A}$ must have full column rank ($\dim E = n$).

The next lemma captures an instrumental geometric property of $\mathcal{K}$ [2].

Lemma 2. *The open cone $\mathcal{K}$ intersects the subspace E = Range($\mathcal{A}$) if and only if $E^\perp \cap \bar{\mathcal{K}} = \{0\}$ (here $E^\perp$ denotes the orthogonal complement of E for the Frobenius inner product). Consequently, strict feasibility is characterized by $E \cap \mathcal{K} \neq \emptyset$ while the absence of strict feasibility corresponds to $E^\perp \cap \bar{\mathcal{K}} \neq \{0\}$.*

Proof. This lemma follows from a general result on convex sets. Specifically, if $E \cap \mathcal{K} = \emptyset$ then $\mathcal{K}$ and E can be separated by an hyperplane of equation $\langle S, X\rangle_{\text{Fro}} = 0$ (with S fixed and nonzero) such that

$$\langle S, X\rangle_{\text{Fro}} > 0 \quad \text{for all } X \in \mathcal{K}, \tag{13}$$

$$\langle S, X\rangle_{\text{Fro}} \leqslant 0 \quad \text{for all } X \in E. \tag{14}$$

From (13) it is immediate that $S \in \bar{\mathcal{K}}$. Now, E being a linear subspace, (14) is only possible if in fact $\langle S, X\rangle_{\text{Fro}} = 0$ for all $X \in E$, i.e., if $S \in E^\perp$. Consequently $S \in E^\perp \cap \bar{\mathcal{K}}$.

The converse is proved by contradiction. Suppose that there exist $X \in E \cap \mathcal{K}$ and a nonzero $Y \in E^\perp \cap \bar{\mathcal{K}}$. Then $\langle X, Y\rangle_{\text{Fro}} = 0$ because $X \in E$ and $Y \in E^\perp$, while $\langle X, Y\rangle_{\text{Fro}} > 0$ because $X \in \mathcal{K}$, $Y \in \bar{\mathcal{K}}$, and $Y \neq 0$. This is clearly a contradiction. □

Remark 3. The matrix S characterizing the separating hyperplane in the proof above is the slack variable for the following dual problem of (11):

> Find S in the intersection of $E^\perp$ and $\bar{\mathcal{K}}$.

The primal problem is strictly feasible if and only if $S = 0$ is the only solution of the dual problem.

A two-dimensional interpretation of Lemma 2 is that the angle at the vertex of the cone $\mathcal{K}$ is exactly 90° since $\langle X, Y\rangle_{\text{Fro}} \geq 0$ for all nonzero $X, Y \in \bar{\mathcal{K}}$ with possible equality if both X, Y are on the boundary (see Fig. 2). For tractability in the interior-point framework, we rule out the limit case where $\mathcal{A}x \geq 0$ admits nonzero solutions x, yet is not strictly feasible. This corresponds to the case when both $E \cap \mathcal{K}$ and $E^{\perp} \cap \mathcal{K}$ are empty.

3.1. The Projective algorithm

To find a strictly feasible vector x, the Projective Method relies on the following simple strategy. Given some X in the open cone $\mathcal{K}$, test whether the Dikin ellipsoid centered at X intersects E. If it does, this provides a strictly feasible point since $\Omega(X) \subset \mathcal{K}$. Otherwise, update X to increase the "chances" that this intersection be nonempty. More precisely, the Projective algorithm proceeds as follows.

Algorithm 4 (*Projective algorithm for strict feasibility problems*).
Given an arbitrary initial point $X_0 \in \mathcal{K}$ (e.g., $X_0 = I$), generate a sequence of positive definite matrices X_k as follows. To update X_k into X_{k+1},

Step 1. Compute the orthogonal projection X_k^+ of X_k onto E in the metric $\langle ., . \rangle_{X_k^{-1}}$.

Step 2. If $X_k^+ > 0$, terminate since then $X_k^+ \in \mathcal{K} \cap E$. A strictly feasible point $x \in \mathbb{R}^n$ is then obtained by solving

$$\mathcal{A}x = X_k^+.$$

Step 3. Otherwise, update X_k to X_{k+1} via the formula

$$X_{k+1}^{-1} = X_k^{-1} - \gamma_k X_k^{-1}(X_k^+ - X_k)X_k^{-1}, \tag{15}$$

where the step size γ_k is selected to make X_{k+1}^{-1} positive definite and "larger" than X_k^{-1} by some fixed factor. Specifically, γ_k should be chosen such that

$$\text{Det}(X_{k+1}^{-1}) \geq \kappa \,\text{Det}(X_k^{-1}), \tag{16}$$

where $\kappa > 1$ is a fixed number (both iteration and problem independent).

The key of the Projective Method is of course the existence of a step size γ_k that enforces (16) whenever $\Omega(X_k) \cap E = \emptyset$. This is established in the next subsection and for now we restrict our attention to the convergence mechanism of Algorithm 4.

We first give an intuitive geometric explanation of the convergence mechanism. Assume that $\mathcal{A}x > 0$ is strictly feasible, i.e., that $E^{\perp} \cap \bar{\mathcal{K}} = \{0\}$ as shown in Lemma 2. By construction, the size of X_k^{-1} grows exponentially with the number of iterations since, from (16),

$$\text{Det}(X_k^{-1}) \geq \kappa^k \,\text{Det}(X_0^{-1}), \quad \kappa > 1. \tag{17}$$

Meanwhile, the updating direction

$$\zeta(X_k) := X_k^{-1}(X_k^+ - X_k)X_k^{-1} \tag{18}$$

remains orthogonal to E in virtue of (8) and of the discussion in Section 2.3. Consequently, the sequence X_k^{-1} moves toward infinity *parallel* to $E^\perp$ as the algorithm progresses. But since $E^\perp \cap \mathcal{K} = \{0\}$, no sequence of positive definite matrices can go to infinity along $E^\perp$ without leaving $\mathcal{K}$. This is due to the geometry of the problem as illustrated by Fig. 2, and in particular to the geometry of the cone $\mathcal{K}$ (see Lemma 2). As a result, termination must occur in finitely many iterations, the number of iterations depending only on (1) the growth rate κ, (2) the position of $\mathcal{K}$ with respect to $E^\perp$, and (3) the initial point X_0.

More rigorously, assume strict feasibility, consider a strictly feasible vector $x_f \in \mathbb{R}^n$, and let $X_f := \mathcal{A}x_f \in E \cap \mathcal{K}$. From (17), we have

$$\log\mathrm{Det}(X_k^{-1}) \geqslant k\log\kappa + \log\mathrm{Det}(X_0^{-1})$$

while the arithmetic-geometric mean inequality applied to the eigenvalues of X_k^{-1} yields

$$\log\left(\frac{\mathrm{Tr}(X_k^{-1})}{m}\right) \geqslant \frac{\log(\mathrm{Det}(X_k^{-1}))}{m}.$$

Moreover, $\mathrm{Tr}(X_k^{-1}X_f) \geqslant \alpha\,\mathrm{Tr}(X_k^{-1})$ for some $\alpha > 0$ since $X_f > 0$. The combination of these three inequalities shows the existence of some positive constant τ such that

$$\mathrm{Tr}(X_k^{-1}X_f) \geqslant \tau e^{k\log(\kappa)/m}. \tag{19}$$

Now, since X_k^{-1} is always updated along directions $\zeta(X_k)$ orthogonal to E for $\langle .,. \rangle_{\mathrm{Fro}}$, we have at all iterations k

$$\mathrm{Tr}(X_k^{-1}X_f) = \mathrm{Tr}(X_0^{-1}X_f). \tag{20}$$

Hence

$$\mathrm{Tr}(X_0^{-1}X_f) \geqslant \tau e^{k\log(\kappa)/m} \tag{21}$$

must hold as long as X_k^+ fails to be positive definite. Recalling that $\log\kappa > 0$ since $\kappa > 1$, and observing that the left-hand side is independent of k, this shows that X_k^+ must become positive definite after a finite number of iterations k. In other words, the algorithm will terminate in finitely many steps, the polynomial-time complexity resulting from (21) (see Section 3.3 for details).

In case of infeasibility, by contrast, the algorithm will in principle run forever. In practice, we force termination when the size of X_k^{-1} exceeds some tolerance. Recalling that X_k approaches E, a large norm in X_k^{-1} indicates that for all feasible vectors x (if any), $\mathcal{A}x$ is nearly singular. In other words, problems where this occurs are only marginally feasible or even unfeasible up to rounding errors.

The convergence mechanism is illustrated in Figs. 2 and 3. Note that these two-dimensional pictures are by essence simplistic since nonscalar problems are at least

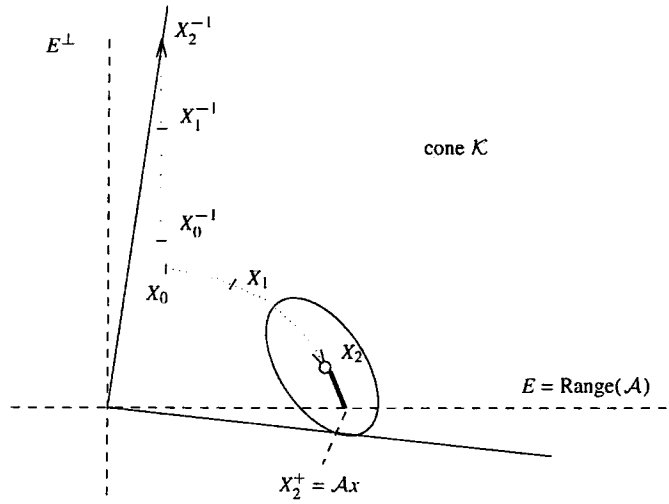

Fig. 2. Convergence mechanism: feasible problem.

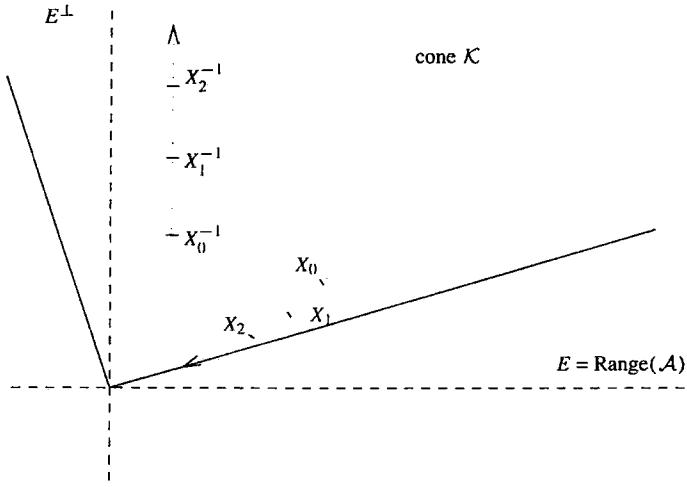

Fig. 3. Convergence mechanism: infeasible problem.

three-dimensional and the geometry of $\mathcal{K}$ is fairly complex. Through these figures we only mean to give insight into the workings of the algorithm, not into the problem geometry. Note that the algorithm actually works toward proving infeasibility, and solves the feasibility problem only by failing to do so.

Computing the orthogonal projection X_k^+ amounts to solving the least-squares problem:

$$\min_{x\in\mathbb{R}^n} \|\mathcal{A}x - X_k\|_{X_k^{-1}}. \tag{22}$$

Even though this task turns out to be most expensive in terms of arithmetic operations, it poses no theoretical nor practical difficulty (see Section 5.2 for more details). Con-

sequently, the main challenge is to find a scalar γ_k that enforces the growth rate (16). This central issue is now addressed in detail.

3.2. Determination of the step size γ_k

Now that the convergence mechanism is understood, we are left with proving that X_k can be updated in such a way that $\mathrm{Det}(X_k^{-1})$ grows by at least some fixed factor $\kappa > 1$. Recall that this updating is performed as long as the orthogonal projection X_k^+ of X_k onto E (for $\langle .,. \rangle_{X_k^{-1}}$) fails to be positive definite. Throughout this subsection, we therefore assume that X_k^+ is not positive definite. As a by-product, we know that the Dikin ellipsoid $\Omega(X_k)$ does not intersect E. Indeed, if $Y \in \Omega(X) \cap E$, then $\|Y - X_k\|_{X_k^{-1}} < 1$ by definition and $\|X_k^+ - X_k\|_{X_k^{-1}} \leqslant \|Y - X_k\|_{X_k^{-1}}$ since X_k^+ is the point of E closest to X_k. As a result, X_k^+ must be in $\Omega(X_k)$ whence $X_k^+ > 0$ by Lemma 1, a contradiction. That $\Omega(X_k) \cap E = \emptyset$ proves critical in the subsequent derivation of an adequate step size γ_k.

The shorthand notation X and X^+ is used in place of X_k and X_k^+ for the rest of the section. Define

$$\zeta(X) := X^{-1}(X^+ - X)X^{-1}. \tag{23}$$

Clearly

$$\rho := \|X^+ - X\|_{X^{-1}} \geqslant 1 \tag{24}$$

since $X^+ \notin \Omega(X)$. To derive an adequate γ_k, it is convenient to work with the barrier $F(x) = -\log\mathrm{Det}\, X^{-1}$ rather than with $\mathrm{Det}\, X^{-1}$. Our goal is then to find $\gamma \in \mathbb{R}$ such that $X^{-1} - \gamma\zeta(X) > 0$ and

$$F(X^{-1}) - F(X^{-1} - \gamma\zeta(X)) \geqslant \theta > 0 \tag{25}$$

for some θ independent of X, or equivalently to show that F can be decreased by some fixed amount in the direction $-\zeta(X)$. This naturally leads to studying the function

$$\pi(\gamma) := F(X^{-1}) - F(X^{-1} - \gamma\zeta(X)) = \log\mathrm{Det}(I - \gamma\psi), \tag{26}$$

where

$$\psi := X^{1/2}\zeta(X)X^{1/2} = X^{-1/2}(X^+ - X)X^{-1/2}. \tag{27}$$

Two important properties of the symmetric matrix ψ are stated in the next lemma.

Lemma 5. *Consider $X > 0$ such that $\Omega(X) \cap E = \emptyset$ and let $\rho := \|X^+ - X\|_{X^{-1}}$. With this notation, the matrix ψ defined by (27) satisfies*

$$\mathrm{Tr}(\psi^2) = \rho^2, \tag{28}$$

$$\mathrm{Tr}(\psi) = -\rho^2. \tag{29}$$

Proof. See Appendix A. □

Given (28)–(29), $\pi(\gamma)$ can be evaluated as follows. Let $\lambda_1, \ldots, \lambda_m$ denote the real eigenvalues of the symmetric matrix ψ, and introduce

$$\rho_\infty := \max_{1 \leqslant i \leqslant m} |\lambda_i|. \tag{30}$$

Clearly $I - \gamma\psi$ is positive definite whenever

$$0 \leqslant \gamma\rho_\infty < 1. \tag{31}$$

In this range of values, the next lemma provides a lower bound on $\pi(\gamma)$ and establishes the existence of an adequate γ.

Lemma 6. *For all γ satisfying* (31),

$$\pi(\gamma) \geqslant \rho^2\{\gamma + \rho_\infty^{-2}(\log(1 - \gamma\rho_\infty) + \gamma\rho_\infty)\} \tag{32}$$

and the right-hand side is maximized for $\gamma^* = 1/(1 + \rho_\infty)$ so that

$$\pi(\gamma) \geqslant \pi^* := \frac{\rho^2}{\rho_\infty^2}(\rho_\infty - \log(1 + \rho_\infty)) \geqslant 1 - \log 2. \tag{33}$$

Proof. See Appendix B. □

Selecting $\gamma := \gamma^* = 1/(1 + \rho_\infty) > 0$ therefore guarantees that

$$\log\mathrm{Det}(X_{k+1}^{-1}) - \log\mathrm{Det}(X_k^{-1}) \geqslant 1 - \log 2$$

or equivalently that

$$\mathrm{Det}(X_{k+1}^{-1}) \geqslant \kappa\,\mathrm{Det}(X_k^{-1})$$

where $\kappa := \exp(1 - \log(2)) = \mathrm{e}/2 \approx 1.36 > 1$.

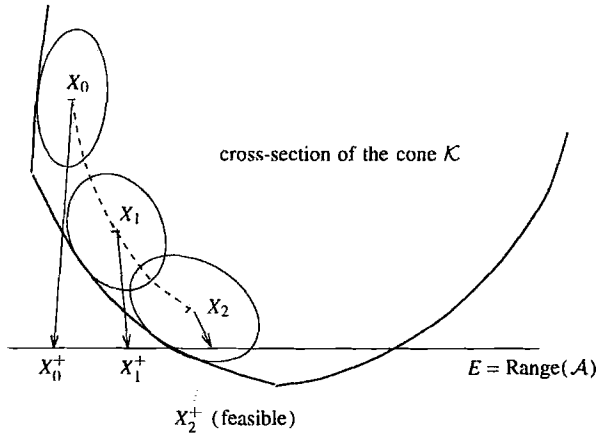

Fig. 4. Updating of X_k.

When implementing this updating, it is more efficient to directly maximize $\pi(\gamma)$ via a line search. To evaluate $\pi(\gamma)$ and its first and second derivatives at a low cost, ψ is first reduced to tridiagonal form via orthogonal similarity. The cost of each evaluation of $\pi(\gamma)$ is then of order $O(m)$. The main advantage of the line search is to generally yield a faster growth rate for $\mathrm{Det}(X_k^{-1})$, hence a faster convergence of the algorithm.

A two-dimensional illustration of the sequence generated by this updating appears in Fig. 4.

3.3. Complexity

Summarizing the previous argument, we come to the following complexity estimate.

Theorem 7. *Assume that the strict feasibility problem* (11) *is solvable, and let* m *denote the total row size of* $\mathcal{A}x$. *Then the Projective Method started at* $X_0 > 0$ *will find a strictly feasible point in a finite number* N *of iterations bounded by*

$$N - 1 \leqslant \frac{m}{1 - \log 2} \inf_{X_f \in E,\, X_f > 0} \log \mathrm{Cond}_{X_0}(X_f)$$

where the condition number $\mathrm{Cond}_{X_0}(X_f)$ *of* X_f *with respect to* X_0 *is defined by*

$$\mathrm{Cond}_{X_0}(X_f) := \frac{\min\{t \mid tX_0 \geqslant X_f\}}{\max\{t \mid tX_0 \leqslant X_f\}} = \frac{\lambda_{\max}(X_0^{-1/2} X_f X_0^{-1/2})}{\lambda_{\min}(X_0^{-1/2} X_f X_0^{-1/2})}.$$

Proof. Assume that the method has not yet terminated after k steps and let

$$t_{\min} = \min\{t \mid tX_0 \geqslant X_f\}, \qquad t_{\max} = \max\{t \mid tX_0 \leqslant X_f\}.$$

As seen above in (20),

$$\mathrm{Tr}(X_{k+1}^{-1} X_f) = \mathrm{Tr}(X_0^{-1} X_f)$$

and consequently

$$t_{\max} \mathrm{Tr}(X_0 X_{k+1}^{-1}) \leqslant t_{\min} \mathrm{Tr}(I) = m t_{\min}. \tag{34}$$

Meanwhile, recall from Section 3.2 that

$$\mathrm{Det}(X_0 X_{k+1}^{-1}) \geqslant \kappa^k$$

where $\log \kappa = 1 - \log 2$. This together with (34) implies that the positive definite symmetric matrix $Y = X_0^{1/2} X_{k+1}^{-1} X_0^{1/2}$ satisfies the inequalities

$$\mathrm{Cond}_{X_0}(X_f) \geqslant \frac{\mathrm{Tr}(Y)}{m}, \qquad \mathrm{Det}(Y) \geqslant \kappa^k.$$

Since $\mathrm{Tr}(Y)/m \geqslant \mathrm{Det}^{1/m}(Y)$, we infer that

$$\log \mathrm{Cond}_{X_0}(X_f) \geqslant \frac{k}{m} \log \kappa \geqslant \frac{k}{m}(1 - \log 2)$$

and the desired upper bound on N readily follows. □

4. Minimization of a linear objective under LMI constraints

This section shows how the Projective Method described in Section 3 can be adapted to the problem

$$\text{Minimize } c^T x \quad \text{subject to } A(x) = \mathcal{A}x + \mathcal{B} \geqslant 0 \tag{35}$$

where $\mathcal{A}x = x_1\mathcal{A}_1 + \cdots + x_n\mathcal{A}_n$. To simplify the forthcoming analysis, we assume that

Solvability assumption. Problem (35) is strictly feasible and the optimal solution set is nonempty.

As in Section 3, we first turn (35) into a homogeneous problem. Consider the auxiliary problem

$$\text{Minimize } \frac{c^T y}{\tau} \text{ over } y, \tau \quad \text{subject to } \mathcal{A}y + \tau\mathcal{B} \geqslant 0, \quad \tau > 0.$$

This problem and (35) are equivalent in the change of variable $x = y/\tau$ and share the same global minimum. Defining

$$\tilde{x} := \begin{pmatrix} x \\ \tau \end{pmatrix}, \quad \tilde{\mathcal{A}}\tilde{x} := \begin{pmatrix} \mathcal{A}x + \tau\mathcal{B} & 0 \\ 0 & \tau \end{pmatrix}, \quad \tilde{c} := \begin{pmatrix} c \\ 0 \end{pmatrix}, \quad \tilde{d} := \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

we can therefore replace the original problem by the homogeneous problem:

$$\text{Minimize } \frac{\tilde{c}^T\tilde{x}}{\tilde{d}^T\tilde{x}} \quad \text{subject to } \tilde{\mathcal{A}}\tilde{x} \geqslant 0, \quad \tilde{d}^T\tilde{x} \neq 0. \tag{36}$$

For the sake of clarity, all tildes are dropped in the sequel, bearing in mind that we are dealing with the transformed problem (36). As before, E denotes the range of the linear map $\mathcal{A}$.

Remark 8. Note that in the homogeneous reformulation introduced above, $\tilde{d}^T\tilde{x} > 0$ whenever $\tilde{\mathcal{A}}\tilde{x} > 0$, that is, whenever x is strictly feasible.

The first task consists of finding a feasible vector x. The objective

$$\Theta(x) := \frac{c^T x}{d^T x} \tag{37}$$

plays no role at this stage and Algorithm 4 can be applied until a strictly feasible vector x_0 is found; without loss of generality, we can assume that the τ-component of this vector is 1. Let $X_0 = \mathcal{A}x_0 > 0$ denote the corresponding value of the LMI, and let $\theta_0 := c^T x_0/d^T x_0$. The level sets of the function Θ are the hyperplanes

$$\{x \in \mathbb{R}^n \mid \Theta(x) = \theta\} = \{x \in \mathbb{R}^n \mid (c - \theta d)^T x = 0\}$$

and they are mapped by $\mathcal{A}$ to

$$E(\theta) := \{\mathcal{A}x \in E \mid (c - \theta d)^T x = 0\}. \tag{38}$$

We are now ready to outline the Projective Method for linear objective minimization.

Algorithm 9 (*Projective algorithm for linear objective minimization*).
Given an initial strictly feasible $X_0 = \mathcal{A}x_0 \in \mathcal{K} \cap E$ computed by Algorithm 4, generate a sequence of matrices $X_k > 0$ and a sequence θ_k^* of objective values at strictly feasible solutions x_k^* as follows. To update X_k, θ_k^*, x_k^* to $X_{k+1}, \theta_{k+1}^*, x_{k+1}^*$:

Step 1. Compute the orthogonal projection $X_k^+ = \mathcal{A}x_k$ of X_k onto E (for the metric $\langle .,. \rangle_{X_k^{-1}}$) and check its positive definiteness. If $X_k^+ > 0$, call the step productive and go to Step 2. Otherwise, call the step unproductive, set

$$\theta_k^* := \theta_{k-1}^*, \quad x_k^* := x_{k-1}^*, \quad Y_k := X_k^+ - X_k,$$

and go directly to Step 3.

Step 2. Decrease θ starting from $\theta_k = c^T x_k / d^T x_k$ until the orthogonal projection $X_k^+(\theta)$ of X_k onto $E(\theta)$ (still in the metric $\langle .,. \rangle_{X_k^{-1}}$) is about to leave $\mathcal{K}$. More precisely, find $\theta \leqslant \theta_k$ such that

$$\|X_k - X_k^+(\theta)\|_{X_k^{-1}} \geqslant 0.99 \quad \text{and} \quad X_k^+(\theta) > 0. \tag{39}$$

Let θ_k^* be the resulting value, x_k^* be the strictly feasible vector such that $X_k^+(\theta_k^*) = \mathcal{A}x_k^*$, and set

$$Y_k := X_k^+(\theta_k^*) - X_k.$$

Step 3. Update X_k to X_{k+1} according to

$$X_{k+1}^{-1} = X_k^{-1} - \gamma_k X_k^{-1} Y_k X_k^{-1}, \tag{40}$$

where the step size γ_k is selected such that $X_{k+1}^{-1} > 0$ and

$$\mathrm{Det}(X_{k+1}^{-1}) \geqslant \kappa \, \mathrm{Det}(X_k^{-1}) \tag{41}$$

for some fixed $\kappa > 1$. Go back to Step 1.

Note that the first iteration is always productive since x_0 is strictly feasible and consequently $X_0 = X_0^+ \in E$. Hence x_k^* is well-defined for all $k \geqslant 1$. Moreover, x_k^* is by construction a strictly feasible solution of $\mathcal{A}x \geqslant 0$ with corresponding objective value $\Theta(x_k^*) = \theta_k^*$.

Remark 10. Note that the objective remains unchanged when the iteration is unproductive (see Step 1). Thus the method does not necessarily improve the objective at each iteration, even though it still achieves some progress of its own in unproductive steps.

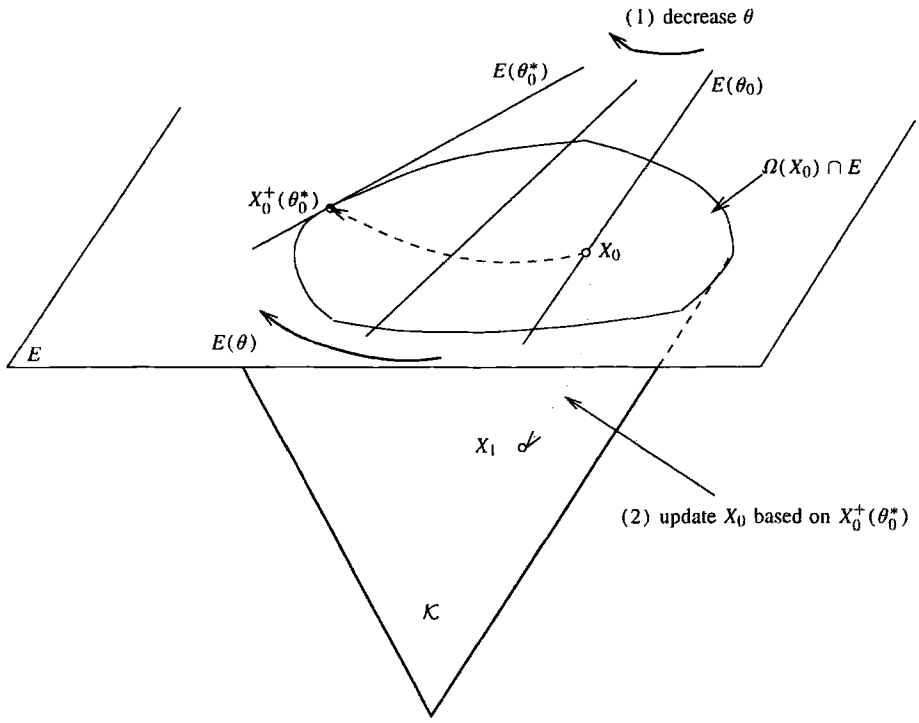

Fig. 5. Progress of the algorithm: first step.

Before discussing this modified algorithm and its convergence properties, we give a graphical illustration of the algorithm progress. The first step is depicted by Fig. 5.

Given the initial feasible solution $X_0 = \mathcal{A}x_0$ and the corresponding objective value θ_0, the objective value θ is decreased in a way ensuring (39) (see Section 4.2). This yields a current best value θ_0^* of the objective as well as a matrix $X_0^+(\theta_0^*)$ in $E(\theta_0^*) \cap \Omega(X_0)$. The latter is used to update X_0 according to Step 3.

The following steps proceed in a similar fashion (see Fig. 6). First the matrix X_k is updated until its projection $X_k^+ = \mathcal{A}x_k$ onto E regains positive definiteness. The value $\theta_k = \Theta(x_k)$ is then a feasible value of the objective, and we can decrease θ down from the value θ_k until (39) is satisfied. Denoting by θ_k^* the resulting objective value, the projection of $X_k^+(\theta_k^*)$ of X_k onto $E(\theta_k^*)$ is used to update X_k to X_{k+1}.

4.1. Step size γ_k

First consider Step 3 and the determination of γ_k such that (41) holds. In unproductive steps, the situation is completely similar to that in Algorithm 4, and from Section 3.2 there exists an adequate γ_k that results in $\kappa = \exp(1 - \log 2)$.

Next consider the case of a productive step. Here again the set-up parallels that of Algorithm 4 with E replaced by $E(\theta_k^*)$. By picking θ_k^* such that $X_k^+(\theta_k^*)$ is outside the ellipsoid

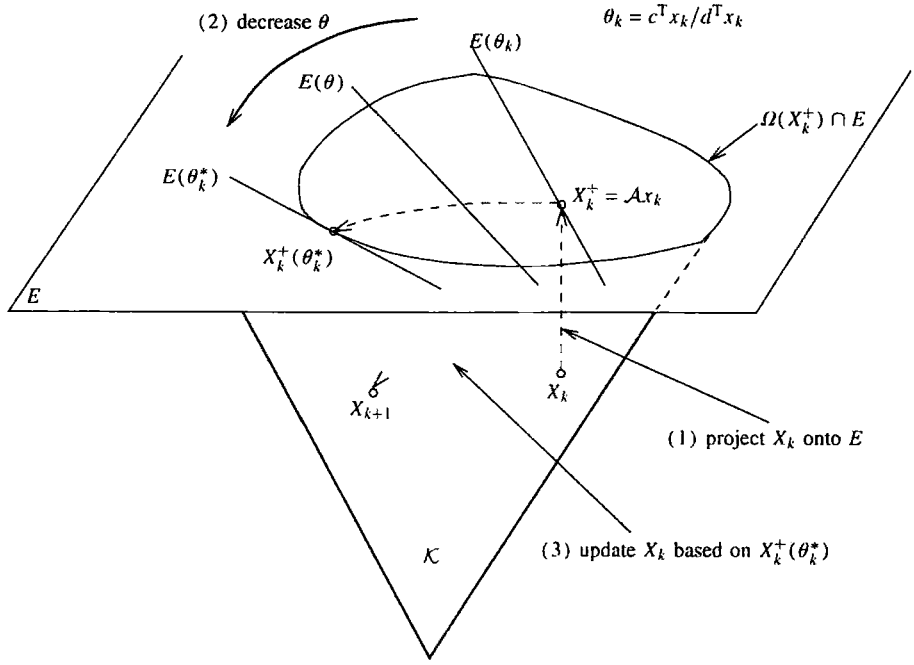

Fig. 6. Progress of the algorithm: following steps.

$$\Omega_{0.99}(X_k) = \{Y \mid \|Y - X_k\|_{X_k^{-1}} < 0.99\}, \tag{42}$$

the existence of some adequate γ_k is established as in Section 3.2, except that the resulting κ is now $\exp(0.99 - \log(1.99))$ instead of $\exp(1 - \log(2))$. This difference is immaterial since $\exp(0.99 - \log(1.99)) > 1$ as well. Hence Step 3 causes no additional difficulty.

4.2. Computation of θ_k^*

Next we turn to the computation of θ_k^* in productive steps. Since $\mathcal{A}$ is a bijection from $\mathbb{R}^n$ to E, there exist two matrices $C_k, D_k \in E$ such that for all $X = \mathcal{A}x \in E$,

$$\langle C_k, X\rangle_{X_k^{-1}} = c^T x, \qquad \langle D_k, X\rangle_{X_k^{-1}} = d^T x. \tag{43}$$

Consequently, $E(\theta)$ defined in (38) is an hyperplane of E with normal vector $C_k - \theta D_k$ and equation

$$E(\theta) := \{X \in E \mid \langle C_k - \theta D_k, X\rangle_S = 0\}, \qquad S := X_k^{-1}. \tag{44}$$

Using Pythagoras theorem in the triangle $(X_k, X_k^+, X_k^+(\theta))$ and the fact that $X_k^+(\theta)$ is the orthogonal projection of X_k^+ onto the hyperplane $E(\theta)$ with normal vector $C_k - \theta D_k$, the squared distance from X_k to $X_k^+(\theta)$ is given by

$$\begin{aligned}\delta^2(\theta) &:= \|X_k - X_k^+(\theta)\|_S^2 \\ &= \|X_k - X_k^+\|_S^2 + \|X_k^+ - X_k^+(\theta)\|_S^2 \\ &= \|X_k - X_k^+\|_S^2 + \frac{\langle C_k - \theta D_k, X_k^+\rangle_S^2}{\|C_k - \theta D_k\|_S^2}.\end{aligned}$$

From the previous subsection, we must find θ_k^* such that

(i) $X_k^+(\theta_k^*)$ is positive definite;

(ii) $X_k^+(\theta_k^*)$ lies outside of the ellipsoid $\Omega_{0.99}(X_k)$ defined in (42).

Since the step in question is productive, $X_k^+ = X_k^+(\theta_k)$ is positive definite. If X_k^+ already lies outside $\Omega_{0.99}(X_k)$, simply take $\theta_k^* = \theta_k$. Otherwise,

- $\|X_k - X_k^+\|_S < 0.99$ implies that

$$\delta(\theta_k) < 0.99 \tag{45}$$

since $\langle C_k - \theta_k D_k, X_k^+\rangle_S = (c - \theta_k d)^T x_k = 0$ by definition of θ_k.

- $x_k(\theta)$ must become infeasible as θ approaches $-\infty$. Otherwise, the objective would be unbounded from below since the objective value at $x_k(\theta)$ is exactly θ, and this would contradict our Solvability Assumption. As a result, $X_k^+(\theta) = \mathcal{A}x_k(\theta)$ must lie outside of the Dikin ellipsoid $\Omega(X_k)$ for small enough θ, whence

$$\lim_{\theta \to -\infty} \delta(\theta) \geqslant 1. \tag{46}$$

By continuity, (45)–(46) prove the existence of $\theta_k^* \leqslant \theta_k$ such that $\delta(\theta_k^*) = 0.99$. Observing that $\delta^2(\theta)$ is the ratio of two quadratic expressions, this value θ_k^* is given by a simple explicit formula. Note that by construction $X_k^+(\theta_k^*)$ belongs to the boundary of the ellipsoid $\Omega_{0.99}(X_k)$ and is therefore positive definite (see Lemma 1).

Remark 11. In the actual implementation of the algorithm, the value θ_k^* derived above is further decreased by a line search down to the value of θ at which $X_k^+(\theta)$ leaves the cone $\mathcal{K}$.

4.3. Convergence and complexity

Denote by θ_{opt} the optimal value of the objective $\Theta(x)$ and by x_{opt} an optimal solution of the problem (that is, such that $A(x_{\text{opt}}) \geqslant 0$ and $\Theta(x_{\text{opt}}) = \theta_{\text{opt}}$); without loss of generality, we assume that the τ-component of x_{opt} is 1. What is meant by polynomial-time convergence in the context of linear objective minimization? We say that the algorithm converges in polynomial time if for some fixed "scale factor" $R > 0$ and for any $\varepsilon \in (0, 1)$, the objective value $\theta_{\text{opt}} + R\varepsilon$ is attained within a finite number N_ε of iterations bounded from above by

$$N_\varepsilon \leqslant O(\pi(m, n)) \log(C/\varepsilon) \tag{47}$$

where m is the size of $A(x)$, n the number of scalar variables x_i, $\pi(.,.)$ is a polynomial, and C is a data-dependent factor. For our purpose, it is convenient to take $R := \Theta(x_0) - \theta_{\text{opt}}$ and $C := \min\{t \mid tX_0 \geqslant X_{\text{opt}}\}$ where $X_0 = \mathcal{A}x_0 > 0$ and $X_{\text{opt}} = \mathcal{A}x_{\text{opt}} \geqslant 0$.

To prove polynomial-time convergence, fix $\varepsilon \in (0,1)$ and introduce

$$x_\varepsilon = (1-\varepsilon)x_{\text{opt}} + \varepsilon x_0, \qquad X_\varepsilon = \mathcal{A}x_\varepsilon. \tag{48}$$

Clearly x_ε is a strictly feasible vector such that

$$\Theta(x_\varepsilon) \leqslant \theta_{\text{opt}} + R\varepsilon \tag{49}$$

(recall that the τ-components of both x_0 and x_{opt} are equal to 1, so that $\Theta(x) = c^{\mathrm{T}}x$ for $x = x_0, x_{\text{opt}}, x_\varepsilon$). Let N_ε be the number of iterations performed until $\theta_k^* \leqslant \Theta(x_\varepsilon)$. To bound N_ε from above, we use an argument similar to that of Section 3.3.

In Algorithm 4, the convergence proof relied critically on the following fact: given any strictly feasible $X = \mathcal{A}x \in \mathcal{K}$, the quantity $\mathrm{Tr}(X_k^{-1}X)$ remained constant throughout the updating of X_k, that is, $\mathrm{Tr}(X_0^{-1}X) = \mathrm{Tr}(X_k^{-1}X)$ for all k. Even though this property no longer holds for the modified algorithm, the next lemma shows that for X_ε introduced above, we have $\mathrm{Tr}(X_{k+1}^{-1}X_\varepsilon) \leqslant \mathrm{Tr}(X_k^{-1}X_\varepsilon)$ as long as $\theta_k^* > \Theta(x_\varepsilon)$, that is, for all $k \leqslant N_\varepsilon$. As a result, $\mathrm{Tr}(X_\varepsilon X_k^{-1})$ is always bounded from above by $\mathrm{Tr}(X_\varepsilon X_0^{-1})$, which allows us to adapt the derivation of Section 3.3 to derive an upper bound for N_ε.

Lemma 12. *Let x_ε and $X_\varepsilon = \mathcal{A}x_\varepsilon$ be defined by* (48). *Then*

$$\mathrm{Tr}(X_{k+1}^{-1}X_\varepsilon) \leqslant \mathrm{Tr}(X_k^{-1}X_\varepsilon) \tag{50}$$

holds as long as $\theta_k^* > \Theta(x_\varepsilon)$.

Proof. Consider some iteration $k < N_\varepsilon$ where $\theta_k^* > \Theta(x_\varepsilon)$. If this iteration is unproductive, the updating is identical to that of Algorithm 4 and consequently $\mathrm{Tr}(X_{k+1}^{-1}X_\varepsilon) = \mathrm{Tr}(X_k^{-1}X_\varepsilon)$. Assume from now on that the iteration is productive. We first show that

$$X_k^+(\theta_k^*) - X_k^+ = -\alpha(C_k - \theta_k^* D_k) \quad \text{with } \alpha \geqslant 0 \tag{51}$$

with C_k, D_k defined by (43). To this end, consider x_k and x_k^* such that

$$X_k^+ = \mathcal{A}x_k, \qquad X_k^+(\theta_k^*) = \mathcal{A}x_k^*.$$

Recalling that $X_k^+(\theta_k^*)$ is the orthogonal projection of X_k^+ onto $E(\theta_k^*)$, and that $E(\theta_k^*)$ is an hyperplane of E with normal vector $C_k - \theta_k^* D_k$, the vector $X_k^+(\theta_k^*) - X_k^+$ must be collinear to $C_k - \theta_k^* D_k$. In addition,

$$\begin{aligned}\langle C_k - \theta_k^* D_k, X_k^+(\theta_k^*) - X_k^+ \rangle_{X_k^{-1}} &= (c - \theta_k^* d)^{\mathrm{T}} x_k^* - (c - \theta_k^* d)^{\mathrm{T}} x_k \\ &= (\Theta(x_k^*) - \theta_k^*) d^{\mathrm{T}} x_k^* - (\Theta(x_k) - \theta_k^*) d^{\mathrm{T}} x_k \\ &= -(\theta_k - \theta_k^*) d^{\mathrm{T}} x_k \leqslant 0\end{aligned}$$

since $\Theta(x_k) = \theta_k \geqslant \theta_k^* = \Theta(x_k^*)$ and $d^{\mathrm{T}}x_k > 0$ from $\mathcal{A}x_k > 0$. This establishes (51).

Back to the proof of (50), we have from (40):

$$\begin{aligned}\gamma_k \operatorname{Tr}((X_k^{-1} - X_{k+1}^{-1})X_\varepsilon) &= \operatorname{Tr}(X_k^{-1}(X_k^+(\theta_k^*) - X_k)X_k^{-1}X_\varepsilon)\\ &= \langle X_k^+(\theta_k^*) - X_k, X_\varepsilon\rangle_{X_k^{-1}}\\ &= \langle X_k^+(\theta_k^*) - X_k^+, X_\varepsilon\rangle_{X_k^{-1}} + \langle X_k^+ - X_k, X_\varepsilon\rangle_{X_k^{-1}}\end{aligned}$$

Now,

- $\langle X_k^+ - X_k, X_\varepsilon\rangle_{X_k^{-1}} = 0$ since $X_k^+ - X_k$ is orthogonal to E while $X_\varepsilon \in E$;
- from (51) it follows that

$$\begin{aligned}\langle X_k^+(\theta_k^*) - X_k, X_\varepsilon\rangle_{X_k^{-1}} &= -\alpha\langle C_k - \theta_k^* D_k, X_\varepsilon\rangle_{X_k^{-1}}\\ &= -\alpha(c - \theta_k^* d)^{\mathrm{T}} x_\varepsilon\\ &= -\alpha(\Theta(x_\varepsilon) - \theta_k^*)d^{\mathrm{T}}x_\varepsilon.\end{aligned}$$

Summing up, we have established the identity

$$\gamma_k \operatorname{Tr}((X_k^{-1} - X_{k+1}^{-1})X_\varepsilon) = -\alpha(\Theta(x_\varepsilon) - \theta_k^*)d^{\mathrm{T}}x_\varepsilon.$$

This completes the proof of (50) upon recalling that (1) $\Theta(x_\varepsilon) < \theta_k^*$ whenever $k < N_\varepsilon$, (2) $\gamma_k > 0$, and (3) α and $d^{\mathrm{T}}x_\varepsilon$ are nonnegative scalars. □

The argument used to prove Theorem 7 is readily adapted to derive an upper bound on N_ε from (50). Here X_ε plays the role of X_f and consequently, the resulting upper bound on N_ε will be $1 + m\log(\operatorname{Cond}_{X_0}(X_\varepsilon))/\log\kappa$ with $\log\kappa \geqslant 0.99 - \log 1.99$. Since $X_\varepsilon = (1-\varepsilon)X^* + \varepsilon X_0$ and X_0, X_ε are positive definite, we clearly have $\operatorname{Cond}_{X_0}(X_\varepsilon) \leqslant C/\varepsilon$. This leads to the following result.

Theorem 13. *Let Problem* (35) *satisfy the Solvability Assumption, and let* x_0 *be a strictly feasible point* ($\mathcal{A}x_0 > 0$). *For any* $\varepsilon \in (0,1)$, *the number of iterations performed by Algorithm* 9 *before a feasible solution with objective value less than* $\theta_{\mathrm{opt}} + \varepsilon(\Theta(x_0) - \theta_{\mathrm{opt}})$ *is found does not exceed the quantity*

$$1 + m\frac{\log(C/\varepsilon)}{0.99 - \log 1.99},$$

where

$$C := \min\{t \mid t\mathcal{A}x_0 \geqslant \mathcal{A}x_{\mathrm{opt}}\},$$

and x_{opt} *is the optimal solution of the problem.*

5. Implementation and numerical results

From the previous discussion, solving the least-squares problem

$$\min_{x\in\mathbb{R}^n} \|\mathcal{A}x - X_k\|_{X_k^{-1}} \tag{52}$$

is the main computational effort involved in each iteration of Algorithms 4 and 9. This section shows that for control-oriented applications, much can be gained by exploiting the specific block-matrix structure of each problem. Numerical stability issues near the optimum are also addressed. Finally, experimental data on the running time and computational overhead for a typical control application are reported. The details given next pertain to the implementation of the Projective Method available in the release 1.0 of the *LMI Control Toolbox* [8,9].

5.1. Structured representation of LMIs

Henceforth, the discussion is implicitly specialized to the case of a *single* homogenous LMI constraint

$$\mathcal{A}x = \sum_{i=1}^{n} x_i \mathcal{A}_i > 0. \tag{53}$$

All arguments readily extend to non-homogenous LMIs and to the case of multiple LMI constraints. The canonical representation (53) is generic and without further information on the LMI structure, the LMI is best described by storing the upper triangles of the symmetric matrices $\mathcal{A}_i$. In most control applications however, LMI constraints are heavily structured which renders the canonical representation quite inefficient. Specifically, most control-like LMIs are of the form

$$\mathcal{A}x = \sum_{r=1}^{n_t} [L_r Z_r R_r^{\mathrm{T}} + R_r Z_r^{\mathrm{T}} L_r^{\mathrm{T}}], \tag{54}$$

where L_r, R_r are given matrices and Z_r is a particular instance in a collection $Y_1, \ldots, Y_V$ of *structured* matrix variables, e.g., symmetric, block-diagonal, Toeplitz, etc. Here the decision variables $x_1, \ldots, x_n$ are the free entries of $Y_1, \ldots, Y_V$ (taking structure into account). From now on, (53) is referred to as the "unstructured" representation while (54) is called the "structured" representation.

To illustrate the benefits of the structured representation, we use the following simple example:

$$AYE^{\mathrm{T}} + EYA^{\mathrm{T}} < 0, \tag{55}$$

where $A, E \in \mathbb{R}^{p \times p}$ are given and Y is a symmetric matrix to be determined. This LMI would be called a generalized Lyapunov inequality. Note that this example is only chosen for its tutorial value, and that in the particular case of Lyapunov inequalities, additional properties could be used to further boost efficiency [19]. However, while the speed-ups obtained in [19] are limited to Lyapunov or Riccati inequalities, the benefits of the structured representation and rank-one linear algebra discussed below pertain to the general class of structured problems (54). To clarify the subsequent argument, we summarize the dimensional parameters involved in Problem (55):

- p denotes the number of states in the problem, that is, the row dimension of A.
- $n = \frac{1}{2}p(p+1)$ is the number of decision variables (free entries of Y when accounting for symmetry).
- $M = \frac{1}{2}p(p+1)$ is the dimension of the image space $\mathcal{S}$ where $AYE^{\mathrm{T}} + EYA^{\mathrm{T}}$ takes values.
- n_t denotes the number of terms in LMIs of the form (54). It is typically small compared to n, and $n_t = 1$ for the LMI (55).

In the flop counts and storage estimates given below, we only keep the dominant term so that, e.g., $\frac{1}{2}p(p+1) + 3p$ is approximated by $\frac{1}{2}p^2$.

An immediate benefit of the structured representation is in terms of storage requirement and cost of evaluating $\mathcal{A}x$. Take the example (55). With the unstructured representation (53), we need to store all $\mathcal{A}_i$ which uses $\frac{1}{4}p^4$ storage since there are $n \approx \frac{1}{2}p^2$ decision variables. By comparison, in the structured representation it suffices to store the matrices A and E which uses $2p^2$ storage. Similarly, evaluating $\mathcal{A}x$ in the unstructured case costs $\frac{1}{2}p^4$ flops vs. only $2p^3$ with the structured representation.

Thus, the structured representation is significantly cheaper in terms of storage requirement and evaluation cost. More importantly, the computational burden attached to solving the least-squares problem (52) can also be significantly reduced. This claim is justified next.

5.2. *Rank-one linear algebra*

The main computational effort in the Projective Method, as in any other interior-point method for this type of problem, is the one required to project a given symmetric X onto the range space of $\mathcal{A}$ in the metric $\langle .,. \rangle_S$ where $S > 0$. That is, to compute the solution $x^* \in \mathbb{R}^n$ of the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|\mathcal{A}x - X\|_S.$$

Let

$$S = LL^{\mathrm{T}}$$

be a Cholesky factorization of S. There are two main linear algebra techniques to compute x^*:

- Cholesky-based approach: x^* is the solution of the normal equation

$$Hx^* = q \tag{56}$$

where H and q are the matrix and vector with entries

$$H_{ij} = \mathrm{Tr}(\mathcal{A}_i S \mathcal{A}_j S), \qquad q_i = \mathrm{Tr}(\mathcal{A}_i L X L^{\mathrm{T}}). \tag{57}$$

To solve (56), we compute a Cholesky factorization

$$H = R^{\mathrm{T}}R$$

of the positive definite matrix H and x^* is then obtained by solving two triangular linear systems. Note that H must be assembled as a preliminary to the Cholesky factorization.

- QR-based approach: introducing the linear mapping $\mathcal{L}$ defined on $\mathcal{S}$ by

$$\mathcal{L} : Z \mapsto L^T Z L,$$

the least-squares problem is equivalent to

$$\min_{x \in \mathbb{R}^n} \| \mathcal{L}\mathcal{A}x - \mathcal{L}X_k \|_{\text{Fro}}.$$

To solve this problem with the QR approach, we first compute the $M \times n$ matrix $\mathcal{B}$ associated with the mapping $\mathcal{L}\mathcal{A}$. Then, by an orthogonal transformation Q, we reduce $\mathcal{B}$ to upper-triangular form:

$$Q\mathcal{B} = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad Q \operatorname{vec}(\mathcal{L}X_k) = \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix}.$$

The least-squares solution x^* is then given by solving

$$Rx^* = \zeta_1. \tag{58}$$

Note that the $n \times n$ matrix R is nothing but the Cholesky factor of H, i.e., $H = R^T R$. In the unstructured case, there is no incentive at all for using the Cholesky approach since (1) the cost of assembling H is comparable to the cost of the QR factorization of $\mathcal{L}\mathcal{A}$, and (2) the normal equation (56) is more badly conditioned than (58) since $\kappa(H) = \kappa(R)^2$ [10]. In the case of LMIs with structure (54), however, assembling the Hessian becomes cheap which makes the Cholesky approach very appealing as long as the least-squares problem remains relatively well-conditioned.

To evaluate the speed-up attached with the Cholesky approach, consider again the simple example (55). The cost of the QR approach is easily estimated as follows: $p^5/3$ flops to form the matrix $\mathcal{B}$ (i.e., compute $L^T \mathcal{A}_i L$ for $i = 1, \ldots, n$) and $2Mn^2 = p^6/4$ to perform the actual QR factorization. In comparison, the Cholesky approach costs $n^3/3 = p^6/24$ to factor the Hessian matrix H, to which we must add the cost of assembling H. As mentioned earlier, this cost is comparable to that of the QR factorization of $\mathcal{B}$ in general since $H = \mathcal{B}^T\mathcal{B}$. However, it drops by orders of magnitude for LMIs of the form (54). To see this, denote by $\{e_i\}_{i=1,\ldots,n}$ the standard orthonormal basis of the design space $\mathbb{R}^n$, and consider the term-oriented representation (54). Then H_{ij} can be decomposed as

$$H_{ij} = \sum_{r=1}^{n_t} \sum_{s=1}^{n_t} H_{ij}^{rs},$$

where

$$\begin{aligned} H_{ij}^{rs} &= \operatorname{Tr}\{S(L_r E_i R_r^T + R_r E_i^T L_r^T) S(L_s E_j R_s^T + R_s E_j^T L_s^T)\} \\ &= 2 \operatorname{Tr}\{E_i (R_r^T S L_s) E_j (R_s^T S L_r)\} + 2 \operatorname{Tr}\{E_i^T (L_r^T S L_s) E_j (R_s^T S R_r)\} \end{aligned} \tag{59}$$

Table 1
Relative expense of the Cholesky and QR approaches

Cholesky approach		QR approach
Assembling	Factorization	Total
$O(Np^4)$	$p^6/24$	$Np^6/4$

and E_i, E_j denote the values of Z_r and Z_s for $x = e_i$ and $x = e_j$, respectively. For typical matrix variable structures, E_i and E_j are low-rank matrices, i.e.,

$$E_i = \sum_{h=1}^{\alpha} \varepsilon_{hi}\delta_{hi}^{\mathrm{T}}, \qquad E_j = \sum_{l=1}^{\beta} \varepsilon_{lj}\delta_{lj}^{\mathrm{T}},$$

where α, β are small integers (0, 1, or 2 in most cases) and ε, δ are canonical vectors of the appropriate space.

Using this low-rank decomposition of E_i and E_j, we obtain

$$H_{ij}^{rs} = 2\sum_{h=1}^{\alpha}\sum_{l=1}^{\beta}\left\{[\delta_{hi}^{\mathrm{T}}(R_r^{\mathrm{T}}SL_s)\varepsilon_{lj}] \times [\delta_{lj}^{\mathrm{T}}(R_s^{\mathrm{T}}SL_r)\varepsilon_{hi}]\right.$$

$$\left. + [\varepsilon_{hi}^{\mathrm{T}}(L_r^{\mathrm{T}}SL_s)\varepsilon_{lj}] \times [\delta_{lj}^{\mathrm{T}}(R_s^{\mathrm{T}}SR_r)\delta_{hi}]\right\}. \tag{60}$$

Now, this expression can be evaluated in $O(1)$ flops since

- All matrix products $R_r^{\mathrm{T}}SL_s, R_s^{\mathrm{T}}SL_r, \ldots$ can be computed beforehand (once for all H_{ij}) at a negligible cost.
- The scalar products $\delta_{hi}^{\mathrm{T}}(R_r^{\mathrm{T}}SL_s)\varepsilon_{lj}, \delta_{lj}^{\mathrm{T}}(R_s^{\mathrm{T}}SL_r)\varepsilon_{hi}, \ldots$ simply amount to selecting particular entries of these pre-computed matrices.

As a result, each entry H_{ij} of H can be assembled in $O(n_t^2)$ flops by exploiting the structure, and H is therefore assembled in $O(n_t^2 n^2)$ flops.

We call this assembling scheme *rank-one linear algebra*. In example (55) where $n_t = 1$ and the rank of E_i (value of the matrix variable Y for $x = e_i$) is at most two, the precise cost of assembling H is

- $3p^3$ flops to compute the matrices $E^{\mathrm{T}}SE$, $A^{\mathrm{T}}SA$, and $E^{\mathrm{T}}SA$ (once and for all);
- 8 flops to evaluate each H_{ij} via (60), yielding a total of p^4 flops since there are $p^4/8$ entries to be evaluated.

The overall cost of the Cholesky approach is therefore $p^6/24$. Table 1 summarizes the previous analysis. The flop counts are given for a system of N Lyapunov inequalities of type (55).

These figures are clearly in favor of the Cholesky approach. Note that the cost of QR grows linearly with the number of LMI constraints, while the dominant term in the Cholesky approach is invariant since it only depends on the number of variables in the problem. In the general case of block-matrix LMIs such as

$$\begin{pmatrix} A^{\mathrm{T}}X + XA + C^{\mathrm{T}}C & XB \\ B^{\mathrm{T}}X & -tI \end{pmatrix} < 0,$$

the block structure of the L_r and R_r factors can be further exploited to eliminate all multiplications by zero blocks. Overall, the rank-one assembling scheme is very efficient as long as the average number n_t of terms involving a given decision variable x_i remains small. Finally, note that for an LMI

$$x_1\mathcal{A}_1 + \cdots + x_n\mathcal{A}_n < 0,$$

where $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are unstructured $p \times p$ matrices, the structured representation does no worse than its unstructured counterpart provided that we exploit the scalar nature of the variables $x_1, \ldots, x_n$.

Remark 14. In linear objective minimization, the optimal solution lies on the boundary of the cone $\bar{\mathcal{K}}$. When the optimal objective value ought to be computed with high accuracy, numerical instability may occur in the final stage of the iterative process as we come very close to the boundary of the feasible set. Indeed, $X_k = \mathcal{A}x_k$ is then nearly singular, and $S = X_k^{-1}$ has a large condition number. As a result, the least-squares problem may be badly conditioned, in which case the normal equation becomes difficult to solve accurately since its condition number is roughly the square of that of the the least-squares problem, i.e., the condition number of R. In fact, the Cholesky factorization of H may even fail due to rounding errors. When this happens, there is no alternative but switching to the QR approach to solve the least-squares problem. Fortunately, the QR steps are typically a small proportion of the total number of iterations and are often not needed if a relative accuracy of 1% on the optimal value is acceptable (see Section 5.3 for details).

5.3. Numerical results

The Projective Method as implemented in the *LMI Control Toolbox* has been tested on a wide variety of control-oriented applications. The qualitative conclusions of these experiments are as follows:

- The method is fairly fast when the desired relative accuracy on the optimal value of (35) is not too high, i.e., around 10^{-2}. In particular, it quickly finds a feasible solution and knocks off most of the inaccuracy in the objective.
- The total number of iterations seems to be almost independent of the problem size and typically ranges between 20 and 30 for well-conditioned problems and between 40 and 60 for badly conditioned ones.
- When a high accuracy is required, the need to switch to QR-based linear algebra significantly increases the running times and the final iterations account for most of the computational burden. For large problems with a thousand variables, this may result in relatively large running times (see Table 1).

Typically, the rank-one phase produces solutions within 1–5% of the optimal value. Since such accuracy is generally sufficient in control applications, it can be argued that the method is highly efficient for most practical purposes.

To illustrate these conclusions, we present detailed numerical results for the following LMI problem drawn from H_∞ control (see, e.g., [7]):

$$\begin{aligned}
&\text{Minimize } \gamma \\
&\text{subject to } \mathcal{N}_{12}^{\mathrm{T}} \begin{pmatrix} AR+RA^{\mathrm{T}} & B_1 & RC_1^{\mathrm{T}} \\ B_1^{\mathrm{T}} & -\gamma I & D_{11}^{\mathrm{T}} \\ C_1 R & D_{11} & -\gamma I \end{pmatrix} \mathcal{N}_{12} < 0 \\
&\qquad\qquad \mathcal{N}_{21}^{\mathrm{T}} \begin{pmatrix} A^{\mathrm{T}}S+SA & SB_1 & C_1^{\mathrm{T}} \\ B_1^{\mathrm{T}}S & -\gamma I & D_{11}^{\mathrm{T}} \\ C_1 & D_{11} & -\gamma I \end{pmatrix} \mathcal{N}_{21} < 0 \\
&\qquad\qquad \begin{pmatrix} R & I \\ I & S \end{pmatrix} > 0.
\end{aligned}$$

Here the variables are the two $p \times p$ symmetric matrices R and S and the scalar γ. Accordingly, the dimension of the design space is $n = p(p+1)+1$.

This linear objective minimization problem was solved for various values of the state-space dimension p and with the additional constraint

$$\|x\|_2 \leqslant 10^7.$$

Two experiments were conducted:

(1) Optimization using only rank-one linear algebra.

(2) Optimization with a required relative accuracy of 10^{-4} on the optimal γ.[2]

All problems were randomly generated and can therefore be considered well conditioned. Since such problems can also be solved by direct linear algebra techniques [5], we compared the optimal value obtained by LMI optimization with the Riccati-based optimum to derive the final relative accuracy estimates.

The results of these experiments appear in Table 2. The CPU times are for a DEC Alpha 200 4/166 workstation. The number of states, the total number of scalar variables, and the dimension of the image space S are denoted by p, n, and M, respectively.

Table 3 indicates the distribution of the CPU time (in %) between the various linear algebra tasks. These figures are relative to the second experiment and confirm the high cost of the final few QR-based iterations.

[2] There is an additional built-in test to detect when the required accuracy is achieved which is as follows: at a productive iteration k, when an improved value of the objective θ_k^* is obtained, we compute the matrix $\tilde{X}_k = X_k^+(\theta_k^* - \delta)$, δ being the required absolute accuracy in terms of the objective value, and check whether the matrix $Z_k = X_k - \tilde{X}_k$ is positive semidefinite. If it is the case, then the actual optimal value is $\geqslant \theta' \equiv \theta_k^* - \delta$, since the positive semidefinite matrix $X_k^{-1} Z_k X_k^{-1}$ is orthogonal (in the Frobenius Euclidean structure) to the plane $E(\theta')$ and therefore gives a separator of the cone of positive semidefinite matrices and the plane. Thus, the indicated test provides us with a sufficient condition for detecting that the required accuracy is already achieved.

Table 2
Experimental results

p	n	M	First experiment (rank-one only)			Second experiment (10^{-4})		
			iter.	relative error	CPU time	iter.	relative error	CPU time
8	73	292	18	2×10^{-4}	2"	23	2×10^{-6}	4"
12	157	680	18	2×10^{-4}	7"	24	2×10^{-5}	23"
16	273	1178	18	2×10^{-2}	20"	25	3×10^{-5}	1'57"
20	421	1750	19	6×10^{-3}	1'00"	25	1×10^{-6}	6'20"
24	601	2436	19	2×10^{-3}	2'44"	24	5×10^{-5}	16'00"
28	813	3236	19	2×10^{-3}	7'00"	24	1×10^{-5}	39'45"
32	1057	4242	20	4×10^{-3}	15'00"	26	3×10^{-5}	1h30'00"

Table 3
Distribution of CPU time

p	Assembling H	Cholesky	QR factorization	Other
8	20	2	50	28
12	15	2	69	14
16	8	3	82	7
20	7	5	84	4
24	7	6	83	4
28	5	8	82	5
32	4	7	85	4

6. Conclusions

A first-principle and comprehensive description of the Projective Method for solving LMI problems has been given. As a modern interior-point method, this algorithm has polynomial-time complexity and extensive practical experience confirms its high performance on LMI problems. In the context of control-oriented applications, the Projective Method can be implemented in a highly efficient manner so as to minimize the computational overhead per iteration. In addition, experimental results suggest that the number of iterations grows very slowly with the size of the problem.

Appendix A

Proof of Lemma 5. Let $S := X^{-1}$. The first identity is immediate from the definition of ψ:

$$\begin{aligned}\mathrm{Tr}\{\psi^2\} &= \mathrm{Tr}\{S^{1/2}(X^+ - X)S(X^+ - X)S^{1/2}\} = \mathrm{Tr}\{S(X^+ - X)S(X^+ - X)\} \\ &= \|X^+ - X\|_S^2 =: \rho^2.\end{aligned}$$

To derive the second identity, observe that

$$\begin{aligned}\|X^+ - X\|_S^2 &= \mathrm{Tr}\{S(X^+ - X)S(X^+ - X)\}\\ &= \mathrm{Tr}\{S(X^+ - X)SX^+\} - \mathrm{Tr}\{S(X^+ - X)SX\}\\ &= \langle X^+ - X, X^+\rangle_S - \mathrm{Tr}\{S(X^+ - X)\}.\end{aligned}$$

Now, $\langle X^+ - X, X^+\rangle_S = 0$ since X^+ is the orthogonal projection of X onto E for the inner product $\langle .,.\rangle_S$. Consequently,

$$\rho^2 = -\mathrm{Tr}\{S(X^+ - X)\} = -\mathrm{Tr}\{S^{1/2}(X^+ - X)S^{1/2}\} = -\mathrm{Tr}(\psi). \qquad \square$$

Appendix B

Proof of Lemma 6. Using a series expansion of the log function, we obtain

$$\begin{aligned}\pi(\gamma) &= \log\mathrm{Det}(I - \gamma\psi) = \sum_{i=1}^{m}\log(1-\gamma\lambda_i) = -\sum_{k=1}^{\infty}\sum_{i=1}^{m}\frac{\gamma^k\lambda_i^k}{k}\\ &= -\sum_{i=1}^{m}\gamma\lambda_i - \sum_{k=2}^{\infty}\sum_{i=1}^{m}\frac{\gamma^k\lambda_i^k}{k} = \gamma\rho^2 - \sum_{k=2}^{\infty}\sum_{i=1}^{m}\frac{\gamma^k\lambda_i^k}{k}.\end{aligned} \tag{B.1}$$

In the last identity we used that $\mathrm{Tr}(\psi) = -\rho^2$ (see Lemma 5). Now, for $k \geqslant 2$ we have

$$\left|\sum_{i=1}^{m}\lambda_i^k\right| \leqslant \left(\sum_{i=1}^{m}\lambda_i^2\right)\left(\max_{1\leqslant i\leqslant m}|\lambda_i|\right)^{k-2} = \rho^2\rho_\infty^{k-2}$$

using this time $\mathrm{Tr}(\psi^2) = \rho^2$. From the expression (B.1) of $\pi(\gamma)$, it follows that

$$\pi(\gamma) \geqslant \gamma\rho^2 - \sum_{k=2}^{\infty}\frac{\gamma^k\rho^2\rho_\infty^{k-2}}{k} = \rho^2\left\{\gamma + \frac{1}{\rho_\infty^2}[\log(1-\gamma\rho_\infty) + \gamma\rho_\infty]\right\}$$

which is exactly (32).

Elementary calculus shows that the right-hand side is maximized for $\gamma^* = 1/(1+\rho_\infty)$. Finally, to establish $\pi^* \geqslant 1-\log 2$, first observe that the function $f(t) \equiv t^{-2}(t-\log(1+t))$ is monotonically decreasing for $t > 0$ since

$$f(t) = t^{-2}\int_0^t (t-\tau)(1+\tau)^{-2}\,\mathrm{d}\tau = \int_0^1 (1-s)(1+ts)^{-2}\,\mathrm{d}s.$$

Two cases must now be distinguished:

- If $\rho_\infty \leqslant 1$, then $\pi^* \geqslant \rho^2 f(1) = \rho^2(1-\log 2) \geqslant 1-\log 2$ (recalling that $\rho \geqslant 1$ from (24) since we assumed that $\Omega_1(X) \cap E = \emptyset$).
- Otherwise, we have $\pi^* \geqslant \rho^2\rho_\infty^{-2}(1-\log 2)$ since $t \mapsto t-\log(1+t)$ is monotonically increasing. Now, $\rho \geqslant \rho_\infty$ since from Lemma 5:

$$\rho^2 = \mathrm{Tr}(\psi^2) = \sum_{i=1}^{m} \lambda_i^2 \geqslant \max_{1 \geqslant i \geqslant m} \lambda_i^2 = \rho_\infty^2.$$

Hence $\pi^* \geqslant 1 - \log 2$ also holds in this case. □

References

[1] F. Alizadeh, Optimization over the positive-semidefinite cone: Interior-point methods and combinatorial applications, *SIAM J. on Optimization* 5 (1995).

[2] A. Berman, *Cones, Matrices, and Mathematical Programming*, Lecture Notes in Economics and Mathematical Systems, Vol. 79 (Springer, Berlin, 1973).

[3] S.P. Boyd and L. El Ghaoui, Method of centers for minimizing generalized eigenvalues, *Linear Algebra and its Applications* 188 (1993) 63–111.

[4] S.P. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics (SIAM, Philadelphia, PA, 1994).

[5] J.C. Doyle, K. Glover, P. Khargonekar and B. Francis, State-space solutions to standard H_2 and H_∞ control problems, *IEEE Transactions on Automatic Control* 34 (1989) 831–847.

[6] M.K.H. Fan, A second-order interior point method for solving linear matrix inequality problems, submitted to *SIAM J. on Control and Optimization* (1993).

[7] P. Gahinet and P. Apkarian, A linear matrix Inequality approach to H_∞ control, *Internat. J. Robust and Nonlinear Control* 4 (1994) 421–448.

[8] P. Gahinet, A. Nemirovski, A.J. Laub and M. Chilali, The LMI control toolbox, in: *Proc. Conf. Dec. Contr.* (1994) 2038–2041.

[9] P. Gahinet, A. Nemirovski, A.J. Laub and M. Chilali, *LMI Control Toolbox* (The MathWorks Inc., 1995).

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

[11] F. Jarre, An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices, *SIAM J. on Optimization* 31 (1993).

[12] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* 4 (1984) 373–395.

[13] M. Kojima, S. Shindoh and S. Hara, Interior-point methods for the monotone linear complementarity problem in symmetric matrices, Technical Report B-282, Dept. of Information Sciences, Tokyo Institute of Technology, 1994.

[14] Yu. Nesterov and A. Nemirovski, Polynomial-time barrier methods in convex programming, *Ekonomika i Matem. Metody* 24 (1988), in Russian; translation: *Matekon*.

[15] Yu. Nesterov and A. Nemirovski, An interior-point method for generalized linear-fractional problems, *Mathematical Programming* 69 (1995) 177–204.

[16] Yu. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM Studies in Applied Mathematics (SIAM, Philadelphia, PA, 1994).

[17] M.L. Overton, Large-scale optimization of eigenvalues, *SIAM J. on Optimization* 2 (1992) 88–120.

[18] F. Rendl and H. Wolkowicz, Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem, *Mathematical Programming* 53 (1992) 63–78.

[19] L. Vandenberghe and S. Boyd, Primal–dual potential reduction method for problems involving matrix inequalities, *Mathematical Programming* 69 (1995) 205–236.