

Fast Algorithms for Projection on an Ellipsoid *

Yu-Hong Dai [†]

Numerical Analysis Report NA/220, January 2004

Abstract

Several fast algorithms are proposed for the problem of projecting a point onto a general ellipsoid. To avoid the direct estimation of the spectral radius in the Lin-Han algorithm, we provide the maximal 2-dimensional inside ball algorithm and the sequential 2-dimensional projection algorithm. However, we find that the solution procedure of the former algorithm may tend to some 2-dimensional reduced ellipsoid and the latter algorithm may produce zigzags. Therefore we investigate the hybrid use of the two algorithms. Our numerical experiments show that all the algorithms, even the hybrid algorithms, are suitable for large-scale problems and much faster than the Lin-Han algorithm. Linear convergence of the algorithms is established. Possible extensions of the algorithms are also discussed.

1. Introduction

The problem of projection on a general convex set

$$\begin{aligned} \min \quad & d(\mathbf{a}, \mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{C}, \end{aligned} \tag{1.1}$$

where $d(\cdot, \cdot)$ is some distance function and \mathcal{C} is some convex set in \mathcal{R}^n , is one of the fundamental problems in convex analysis. It is also an important

*This work was supported by the EPSRC in UK (no. GR/R87208/01) and the Chinese NSF grants (no. 10171104 and 40233029).

[†]Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK & LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Box 2719, Beijing 100080, China. Email: dyh@lsec.cc.ac.cn

inertia of projection methods for nonlinear programming, variational inequality problem, *etc.* For example, Birgin, Martínez, and Raydan [1] established efficient spectral projected gradient algorithms for optimization over convex sets. Evidently, the performance of their algorithms is very related to the sub projection algorithm on the convex set. Although the problem (1.1) has been well studied in theory, it is little known about how to solve the problem except when \mathcal{C} is some special set such as a ball, a box, a box with a singly linear constraint (for example see [12, 2]), or an order simplex (for example see [6]).

In this paper, we consider the following problem of projecting a point onto a general ellipsoid

$$\begin{aligned} \min \quad & d(\mathbf{a}, \mathbf{x}) = \|\mathbf{x} - \mathbf{a}\| \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{E} := \{\mathbf{x} \in \mathcal{R}^n : q(\mathbf{x}) \leq \alpha\}, \end{aligned} \quad (1.2)$$

where $\mathbf{a} \in \mathcal{R}^n$ is a point to be projected, $q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + 2\mathbf{b}^T \mathbf{x}$, A is a positive definite matrix in $\mathcal{R}^{n \times n}$ and $\|\cdot\|$ means the 2-norm. Note that the convex set \mathcal{C} can be usually written as

$$\mathcal{C} = \cap_{i=1}^m \{\mathbf{x} \in \mathcal{R}^n : g_i(\mathbf{x}) \leq 0\}, \quad (1.3)$$

where m is some positive integer and $g_i(\mathbf{x}) (i = 1, \dots, m)$ is some concave function in \mathcal{R}^n . Since a suitable local approximation to a nonlinear function is a quadratic function, the problem (1.2) is fundamental in solving the problem (1.1) with \mathcal{C} given by (1.3). If the problem with $m = 1$ is well solved, one can then use the methods in [4] and [7] *etc.* to solve the general problem with any m . The problem (1.2) with $\mathbf{b} = \mathbf{0}$ is also related to the trust region subproblem in nonlinear optimization.

To solve the problem (1.2), Lin and Han [9] proposed a simple and geometric algorithm for the problem (1.2) with $\mathbf{b} = \mathbf{0}$ with attractive convergence properties. Suppose that the current iteration is \mathbf{x}_k that belongs to the boundary of \mathcal{E} . The basic idea of their algorithm is to construct an n -dimensional ball that lies inside the ellipsoid \mathcal{E} and is tangent to the boundary of \mathcal{E} at \mathbf{x}_k , and then take \mathbf{x}_{k+1} to be the intersection of the boundary of \mathcal{E} and the line segment connecting \mathbf{a} and the center of the ball \mathbf{c}_k . Consequently, they have to estimate the spectral radius of A in some way. As analyzed in Section 3, however, a lower estimate to this quantity may deteriorate the performance of the algorithm greatly. Nevertheless, if we consider the choice of \mathbf{x}_{k+1} on the 2-dimensional linear manifold $\mathcal{S}_k = \mathbf{x}_k + \text{Span}\{\mathbf{a} - \mathbf{x}_k, A\mathbf{x}_k + \mathbf{b}\}$, then much faster algorithms can be obtained.

The rest of this paper is organized as follows. In the next section, we present a general framework for all the algorithms considered in this paper. In Section 3, a numerical analysis on the Lin-Han algorithm is provided. Two new algorithms, namely, maximal 2-dimensional inside ball algorithm and sequential

2-dimensional projection algorithm, are proposed in Sections 4 and 5, respectively. In Section 6, we investigate the hybrid use of the two algorithms and propose the simple hybrid projection algorithm and the general hybrid projection algorithm. Linear convergence result is established in Section 7 for the general hybrid projection algorithm, that has the Lin-Han algorithm, maximal 2-dimensional inside ellipsoid algorithm, and sequential 2-dimensional projection algorithm as its special cases. Numerical results with the algorithms are reported in Section 8. Discussion is made in the last section.

2. The General Algorithm

Throughout this paper, we assume that $q(\mathbf{a}) > \alpha$ for otherwise the projection of \mathbf{a} on the ellipsoid \mathcal{E} is itself. We also assume that

$$\alpha > \min\{q(\mathbf{x}) : \mathbf{x} \in \mathcal{R}^n\} = -\mathbf{b}^T A^{-1} \mathbf{b}, \quad (2.1)$$

so that \mathcal{E} exists and is not a singleton. In this section, we describe a general algorithm whose diagram is shown in Figure 1. This algorithm requires a feasible initial point \mathbf{x}_0 and generates a sequence $\{\mathbf{x}_k\} \subset \Omega(\mathcal{E})$, where $\Omega(\mathcal{E})$ is the boundary of \mathcal{E} ,

$$\Omega(\mathcal{E}) = \{\mathbf{x} \in \mathcal{R}^n : q(\mathbf{x}) = \alpha\}. \quad (2.2)$$

Suppose that a feasible point \mathbf{x}_k is obtained at the k -th iteration. Denote $\mathbf{u}_k = \nabla q(\mathbf{x}_k)/2 = A\mathbf{x}_k + \mathbf{b}$. The algorithm calculates an intermediate point \mathbf{c}_k along the negative gradient of q at \mathbf{x}_k :

$$\mathbf{c}_k = \mathbf{x}_k - \gamma_k \mathbf{u}_k, \quad (2.3)$$

where $\gamma_k > 0$ is so chosen that $\mathbf{c}_k \in \mathcal{E}$, namely, $q(\mathbf{c}_k) \leq \alpha$. For any $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$ with $\mathbf{x} \neq \mathbf{y}$, denote by $\mathcal{L}(\mathbf{x}, \mathbf{y})$ the line segment connecting \mathbf{x} and \mathbf{y} ,

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \{\mathbf{x} + \eta(\mathbf{y} - \mathbf{x}) : \eta \in [0, 1]\}. \quad (2.4)$$

The algorithm takes \mathbf{x}_{k+1} as the minimizer of the distance $\|\mathbf{x} - \mathbf{a}\|$ on the set $\mathcal{L}(\mathbf{a}, \mathbf{c}_k) \cap \mathcal{E}$. Equivalently, defining $\mathbf{w}_k = \mathbf{c}_k - \mathbf{a}$, the algorithm calculates

$$\mathbf{x}_{k+1} = \mathbf{a} + \eta_k \mathbf{w}_k, \quad (2.5)$$

where $\eta_k \in (0, 1)$ is such that $\mathbf{x}_{k+1} \in \Omega(\mathcal{E})$. The above procedure is then repeated until some convergence criterion is satisfied.

Let us denote $\mathbf{g}_a = \nabla q(\mathbf{a})/2 = A\mathbf{a} + \mathbf{b}$, the requirement $\mathbf{x}_{k+1} \in \Omega(\mathcal{E})$ asks η_k to satisfy

$$\alpha = q(\mathbf{x}_{k+1}) = q(\mathbf{a} + \eta_k \mathbf{w}_k) = (\mathbf{w}_k^T A \mathbf{w}_k) \eta_k^2 + 2(\mathbf{g}_a^T \mathbf{w}_k) \eta_k + q(\mathbf{a}) := \psi(\eta_k). \quad (2.6)$$

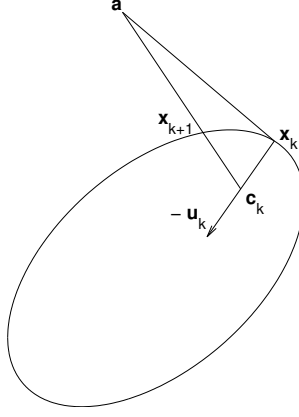


Figure 1. Diagram of The General Algorithm

Notice that $\psi(0) = q(\mathbf{a}) > \alpha$, $\psi(1) = q(\mathbf{a} + \mathbf{w}_k) = q(\mathbf{c}_k) \leq \alpha$, and $\psi(\eta) \rightarrow +\infty$ as $\eta \rightarrow +\infty$. Therefore the quadratic equation $\psi(\eta) = \alpha$ has one root in $(0, 1)$ and one root in $(1, +\infty)$. The smaller one is taken for η_k , namely,

$$\eta_k = -\frac{\mathbf{g}_a^T \mathbf{w}_k}{\mathbf{w}_k^T A \mathbf{w}_k} - \sqrt{\left(\frac{\mathbf{g}_a^T \mathbf{w}_k}{\mathbf{w}_k^T A \mathbf{w}_k}\right)^2 - \frac{q(\mathbf{a}) - \alpha}{\mathbf{w}_k^T A \mathbf{w}_k}}. \quad (2.7)$$

Define $\mathbf{v}_k = \mathbf{a} - \mathbf{x}_k$. The following criterion is used for the termination of the algorithm:

$$1 - \frac{\mathbf{u}_k^T \mathbf{v}_k}{\|\mathbf{u}_k\| \|\mathbf{v}_k\|} \leq \varepsilon, \quad (2.8)$$

where $\varepsilon > 0$ is some tolerance error. Now we provide a detailed description of the general algorithm.

The General Algorithm

1. Given a starting point $\mathbf{x}_0 \in \Omega(\mathcal{E})$ and $\varepsilon > 0$. Set $k := 0$.
2. Calculate γ_k in some way, $\mathbf{u}_k = A\mathbf{x}_k + \mathbf{b}$ and \mathbf{c}_k by (2.3).
3. Calculate $\mathbf{w}_k = \mathbf{c}_k - \mathbf{a}$ and \mathbf{x}_{k+1} by (2.5) and (2.7).
4. If (2.8) does not hold, set $k := k + 1$ and go to step 2.

As will be seen, all the algorithms in this paper are special cases of the above general algorithm, but differ on the choice of γ_k .

3. A Numerical Analysis of the Lin-Han Algorithm

The algorithm by Lin and Han [9], which consists in constructing a ball that lies inside the ellipsoid \mathcal{E} and intersects $\Omega(\mathcal{E})$ at the point \mathbf{x}_k , is a special

ζ	1	0.5	0.2	0.1	0.05	0.02	0.01
#iter	142	284	710	1420	2839	7098	14197

Table 1. Performances of the Lin-Han algorithm with $\gamma_k = 0.01 \zeta$

case of the General Algorithm. More exactly, their algorithm aims to find a positive number γ_k such that the n -dimensional inside ball

$$\mathcal{B}(\gamma_k) = \{\|\mathbf{x} - \mathbf{c}_k\| \leq \gamma_k \|\mathbf{u}_k\| : \mathbf{x} \in \mathcal{R}^n\} \subset \mathcal{E}. \quad (3.1)$$

Consequently, Lin and Han require the choice of γ_k to satisfy the following condition

$$\tau \leq \gamma_k \leq (\rho(A))^{-1}, \quad (3.2)$$

where τ is some small positive constant and $\rho(A)$ is the spectral radius of A .

As analyzed in [9], the first inequality can provide a sufficient decrease of $d(\mathbf{a}, \mathbf{x}_k)$ and hence the global convergence of the algorithm can be established. The function of the second inequality is to guarantee the n -dimensional ball $\mathcal{B}(\gamma_k)$ to lie inside the ellipsoid \mathcal{E} . For this purpose, they need to estimate some matrix norm $\|A\| \geq \rho(A)$ and the 1-norm or ∞ -norm is suggested. As will be shown, the numerical performance of their algorithm heavily relies on the estimation of the spectral radius $\rho(A)$ and an under estimation of this quantity may deteriorate the algorithm greatly.

Consider the following 10-dimensional example:

Example 1. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^{10} : q(\mathbf{x}) := \sum_{i=1}^{10} i^2 x_i^2 = 385\}$, $\mathbf{a} = (a_i)$ with $a_i = 10i^2 + 1$ ($i = 1, \dots, 10$). The initial point is $\mathbf{x}_0 = \sqrt{\frac{385}{q(\mathbf{a})}} \mathbf{a}$. In this example, the projection \mathbf{x}^* of \mathbf{a} on \mathcal{E} is $\mathbf{x}^* = (1, 1, \dots, 1)^T$.

Since in this example $A = \text{diag}(1, 4, \dots, 100)$, we have that $\rho(A) = 100$. We tested the Lin-Han algorithm using $\gamma_k = \zeta (\rho(A))^{-1} = 0.01 \zeta$ with different values of $\zeta \leq 1$. The tolerance error ε in (2.8) is set to 10^{-6} . Table 1 lists the iteration numbers required by the algorithm with different values of ζ .

From Table 1, we see that the number of iterations required by the Lin-Han algorithm is almost linearly dependent on the value ζ . A good estimation of the spectral radius $\rho(A)$ may accelerate the algorithm significantly. This example even suggests that, it is worthwhile to do so before the projection calculations if a good approximation can be obtained with relatively low cost. As will be seen in the following sections, however, this estimation procedure is not necessary and algorithms much faster than the Lin-Han algorithm even with $\gamma_k = (\rho(A))^{-1}$ can be obtained.

4. Maximal 2-Dimensional Inside Ball Algorithm

Our new algorithms are based on the following observation: all the points \mathbf{a} , \mathbf{x}_k , \mathbf{c}_k and \mathbf{x}_{k+1} lie in the 2-dimensional linear manifold

$$\mathcal{S}_k = \{\mathbf{x}_k + (\mathbf{u}_k, \mathbf{v}_k) \mathbf{r} : \mathbf{r} \in \mathcal{R}^2\}, \quad (4.1)$$

where $(\mathbf{u}_k, \mathbf{v}_k)$ stands for a matrix whose columns are \mathbf{u}_k and \mathbf{v}_k . Thus at the k -th iteration we may just consider the 2-dimensional linear manifold \mathcal{S}_k instead of the whole space \mathcal{R}^n .

Define $\mathcal{E}_k = \mathcal{E} \cap \mathcal{S}_k$ to be the *2-dimensional reduced ellipsoid of \mathcal{E}* and $\Omega(\mathcal{E}_k) = \Omega(\mathcal{E}) \cap \mathcal{S}_k$ to be the boundary of \mathcal{E}_k . A direct extension of the Lin-Han algorithm is to construct a 2-dimensional inside ball

$$\mathcal{B}_2(\gamma_k) = \{\|\mathbf{x} - \mathbf{c}_k\| \leq \gamma_k \|\mathbf{u}_k\| : \mathbf{x} \in \mathcal{S}_k\} \subset \mathcal{E}_k. \quad (4.2)$$

In addition, the numerical analysis of Lin-Han algorithm in the previous section suggests that the larger γ_k the more efficient the algorithm. Therefore it is natural for us to choose the maximum 2-dimensional inside ball and propose the following algorithm.

Algorithm 1 (maximal 2-dimensional inside ball algorithm)

At step 2 of the General Algorithm, calculate the maximal γ_k such that (4.2) holds.

Since the dimension of the ellipsoid \mathcal{E}_k is only two, we can directly calculate the radius of the maximum inside ball of \mathcal{E}_k at \mathbf{x}_k and then decide the value of γ_k in the above algorithm. To do this, we orthonormalize the vectors \mathbf{v}_k and \mathbf{u}_k as follows:

$$\mathbf{p}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|}, \quad \mathbf{q}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}, \quad (4.3)$$

where $\mathbf{z}_k = \mathbf{u}_k - \frac{\mathbf{u}_k^T \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} \mathbf{v}_k$. Denote

$$H_k = (\mathbf{p}_k, \mathbf{q}_k) \in \mathcal{R}^{n \times 2}, \quad (4.4)$$

which satisfies $H_k^T H_k = I$. The linear manifold \mathcal{S}_k in (4.1) can be expressed by

$$\mathcal{S}_k = \{\mathbf{x}_k + H_k \mathbf{l} : \mathbf{l} \in \mathcal{R}^2\}. \quad (4.5)$$

Consequently, the 2-dimensional reduced ellipsoid \mathcal{E}_k can be expressed in the vector \mathbf{l} as follows:

$$\mathcal{E}_k^{(l)} = \{\mathbf{l} \in \mathcal{R}^2 : \mathbf{l}^T A_k \mathbf{l} + 2\mathbf{b}_k^T \mathbf{l} \leq 0\}, \quad (4.6)$$

where

$$A_k = H_k^T A H_k = \begin{pmatrix} \frac{\mathbf{v}_k^T A \mathbf{v}_k}{\|\mathbf{v}_k\|^2} & \frac{\mathbf{v}_k^T A \mathbf{z}_k}{\|\mathbf{v}_k\| \|\mathbf{z}_k\|} \\ \frac{\mathbf{z}_k^T A \mathbf{v}_k}{\|\mathbf{v}_k\| \|\mathbf{z}_k\|} & \frac{\mathbf{z}_k^T A \mathbf{z}_k}{\|\mathbf{z}_k\|^2} \end{pmatrix}, \quad \mathbf{b}_k = H_k^T \mathbf{u}_k = \begin{pmatrix} \frac{\mathbf{u}_k^T \mathbf{v}_k}{\|\mathbf{v}_k\|} \\ \frac{\mathbf{u}_k^T \mathbf{z}_k}{\|\mathbf{z}_k\|} \end{pmatrix}. \quad (4.7)$$

At the same time, \mathbf{x}_k corresponds to the origin in the \mathbf{l} subspace. Our problem is then to compute the radius r_k of the maximal inside ball of the ellipsoid $\mathcal{E}_k^{(1)}$ at the origin.

To this aim, for any $t > 0$ we consider the ball

$$\mathcal{B}_2^{(l)}(t) = \{\mathbf{l} \in \mathcal{R}^2 : \|\mathbf{l} + t\mathbf{b}_k\| \leq t\|\mathbf{b}_k\|\}$$

that is tangent with the boundary of $\mathcal{E}_k^{(l)}$ at the origin. For any \mathbf{l} on the boundary of $\mathcal{B}_2^{(l)}(t)$, we have that $\|\mathbf{l} + t\mathbf{b}_k\|^2 = t^2\|\mathbf{b}_k\|^2$ and hence

$$\mathbf{l}^T \mathbf{l} + 2t\mathbf{b}_k^T \mathbf{l} = 0. \quad (4.8)$$

If $t \leq (\rho(A_k))^{-1}$, we can get by this, (4.8) and (4.6) that,

$$\mathbf{l}^T A_k \mathbf{l} + 2\mathbf{b}_k^T \mathbf{l} = \mathbf{l}^T A_k \mathbf{l} - t^{-1} \mathbf{l}^T \mathbf{l} \leq \mathbf{l}^T A_k \mathbf{l} - \rho(A_k) \mathbf{l}^T \mathbf{l} \leq 0,$$

which means $\mathbf{l} \in \mathcal{E}_k^{(l)}$ and hence $r_k \geq (\rho(A_k))^{-1} \|\mathbf{b}_k\|$. On the other hand, for any $t > (\rho(A_k))^{-1}$, consider the point

$$\bar{\mathbf{l}} = -2t(\mathbf{b}_k^T \bar{\mathbf{u}}) \bar{\mathbf{u}},$$

where $\bar{\mathbf{u}}$ is one unit eigenvector of the matrix A_k corresponding to $\rho(A_k)$. By direct check, we know that

$$\bar{\mathbf{l}} \in \mathcal{B}_2^{(l)}(t) \quad \text{but} \quad \bar{\mathbf{l}} \notin \mathcal{E}_k.$$

Hence we also have that $r_k \leq (\rho(A_k))^{-1} \|\mathbf{b}_k\|$. To sum up, $r_k = (\rho(A_k))^{-1} \|\mathbf{b}_k\|$ is exactly the radius of the maximal inside ball of $\mathcal{B}_2^{(1)}(r)$ of \mathcal{E}_k at \mathbf{x}_k .

By direct calculations, we know that the spectral of the matrix A_k is

$$\rho(A_k) = \frac{1}{2} \left[\frac{\mathbf{v}_k^T A \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} + \frac{\mathbf{z}_k^T A \mathbf{z}_k}{\mathbf{z}_k^T \mathbf{z}_k} + \sqrt{\left(\frac{\mathbf{v}_k^T A \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} - \frac{\mathbf{z}_k^T A \mathbf{z}_k}{\mathbf{z}_k^T \mathbf{z}_k} \right)^2 + \frac{4(\mathbf{v}_k^T A \mathbf{z}_k)^2}{\mathbf{v}_k^T \mathbf{v}_k \mathbf{z}_k^T \mathbf{z}_k}} \right]. \quad (4.9)$$

On the other hand, we have by $H_k^T H_k = I$ that $\|\mathbf{b}_k\| = \|H_k^T \mathbf{u}_k\| = \|\mathbf{u}_k\|$. Thus the spectral radius of the maximal 2-dimensional inside ball of \mathcal{E}_k at \mathbf{x}_k is $(\rho(A_k))^{-1} \|\mathbf{u}_k\|$ and the value of γ_k in Algorithm 1 is

$$\gamma_k = (\rho(A_k))^{-1}. \quad (4.10)$$

In the implementation of Algorithm 1, we need not store and compute the vectors \mathbf{p}_k and \mathbf{q}_k since only the value $\rho(A_k)$ is required. We counted that Algorithm 1 requires 1 matrix-vector multiplication and 12 vector-vector operations or scalar-vector multiplications (here note that to calculate $\rho(A_k)$ by (4.9), we can obtain $A\mathbf{z}_k$ and $A\mathbf{v}_k$ by $A\mathbf{u}_k$ and \mathbf{u}_k and hence only require one matrix-vector multiplication, that is $A\mathbf{u}_k$, at each iteration). Comparing with the Lin-Han algorithm, Algorithm 1 requires only 1 more vector-vector operation. However, Algorithm 1 avoids the direct estimate to the spectral radius $\rho(A)$. Even if $\rho(A)$ is available, we may expect that Algorithm 1 is better than the Lin-Han algorithm with $\gamma_k = (\rho(A))^{-1}$ because it follows from $A_k = H_k^T A H_k$ and $H_k^T H_k = I$ that

$$\rho(A_k)^{-1} \geq \rho(A)^{-1}. \quad (4.11)$$

Example 1 in Section 3 has been used for a quick check, and it is found that Algorithm 1 requires only 111 iterations to achieve a solution with the same precision. More numerical comparisons will be provided in Section 8.

5. Sequential 2-Dimensional Projection Algorithm

A numerical drawback of Algorithm 1 is that, even in the case of 2-dimension, if the ellipsoid \mathcal{E} is flat and the point \mathbf{a} to be projected is close to the sharp area, the algorithm may take a large quantity of iterations. Consider the example

Example 2. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^2 : x_1^2 + 10000x_2^2 = 2\}$, $\mathbf{a} = (1, 100.01)^T$. The initial point \mathbf{x}_0 is either $(\sqrt{2}, 0)^T$ or $(-1, 0.01)^T$. The exact projection of \mathbf{a} onto \mathcal{E} is $\mathbf{x}^* = (1, 0.01)^T$.

If $\mathbf{x}_0 = (\sqrt{2}, 0)^T$, Algorithm 1 takes 7361 iterations to reach the stopping condition (2.8) with $\varepsilon = 10^{-6}$. If $\mathbf{x}_0 = (-1, 0.01)^T$, Algorithm 1 requires 12858 iterations to find a satisfactory point. In this example, we have that $\gamma_k \equiv 10^{-4}$. This drawback of Algorithm 1 still exists in the higher dimensional case. Take the 10-dimensional example in Section 3 as an instance. Denoting by M_k the matrix with columns formed by $\mathbf{e}_{k+i} = \frac{\mathbf{x}_{k+i} - \mathbf{x}^*}{\|\mathbf{x}_{k+i} - \mathbf{x}^*\|}$ ($i = 0, 1, 2$), we found that the determinant of $M_k^T M_k$ eventually tends to zero, which means that the solution procedure of Algorithm 1 tends to some 2-dimensional reduced ellipsoid. At the same time, the γ_k tends monotonically increasingly to some value, which is $1.2804e-2$ approximately.

To overcome the above drawback of Algorithm 1, we propose another algorithm that consists in calculating the exact projection of \mathbf{a} onto the 2-dimensional ellipsoid \mathcal{E}_k at each iteration. The following lemma, together with

the invariance of the projection under orthogonal transformations, indicates that the projection onto any 2-dimensional ellipsoid can be obtained via a quartic equation.

Lemma 1 *Consider the 2-dimensional ellipsoid*

$$\mathcal{E}^{(h)} = \{\mathbf{h} \in \mathcal{R}^2 : \mathbf{h}^T D \mathbf{h} \leq \beta\}, \quad (5.1)$$

where $\beta > 0$ and $D = \text{diag}(\lambda_1, \lambda_2)$ with $\lambda_1, \lambda_2 > 0$. For any $\mathbf{h} = (h_1, h_2)^T$ with $\mathbf{h}^T D \mathbf{h} > \beta$, denote by $\mathbf{h}^* = (h_1^*, h_2^*)^T$ the projection of \mathbf{h} onto $\mathcal{E}^{(h)}$. Then

$$h_2^* = \frac{\lambda_1 h_2}{(\lambda_1 - \lambda_2) h_1^* + \lambda_2 h_1} h_1^*, \quad (5.2)$$

where h_1^* satisfies the quartic equation

$$[(\lambda_1 - \lambda_2) h_1^* + \lambda_2 h_1]^2 [\lambda_1 (h_1^*)^2 - \beta] + \lambda_1^2 \lambda_2 h_2^2 (h_1^*)^2 = 0. \quad (5.3)$$

Proof By the Karush-Kuhn-Tucker condition (for example see Fletcher [5]), there exists some $\mu > 0$ such that $\mathbf{h} - \mathbf{h}^* = \mu D \mathbf{h}^*$, or equivalently

$$\begin{cases} h_1 - h_1^* = \mu \lambda_1 h_1^*, \\ h_2 - h_2^* = \mu \lambda_2 h_2^*. \end{cases} \quad (5.4)$$

It follows that

$$\lambda_1 h_1^* (h_2 - h_2^*) = \lambda_2 h_2^* (h_1 - h_1^*), \quad (5.5)$$

which implies the truth of (5.2). In addition, by the feasibility condition,

$$\lambda_1 (h_1^*)^2 + \lambda_2 (h_2^*)^2 = \beta. \quad (5.6)$$

Substituting (5.2) into (5.6), we then know that h_1^* satisfies (5.3). *q.e.d.*

The quartic equation (5.3) can be solved in an analytical way or easily by some numerical methods (in our implementation with MATLAB, we use the function *roots*). From (5.4) and the positiveness of the multiplier μ , we can get that

$$\mu = \frac{h_1 - h_1^*}{\lambda_1 h_1^*} = \frac{h_2 - h_2^*}{\lambda_2 h_2^*} > 0. \quad (5.7)$$

The above relations and (5.2) can help us to pick up the correct value for h_1^* among the four roots of (5.3). The h_2^* is then determined by (5.2).

Note that the computation amount of projecting a point onto a 2-dimensional ellipsoid is negligible when n is relatively large. We propose the following algorithm for projecting onto an n -dimensional ellipsoid.

Algorithm 2 (sequential 2-dimensional projection algorithm)

At the k -th iteration, having $\mathbf{x}_k \in \Omega(\mathcal{E})$ we take the projection of \mathbf{a} on the 2-dimensional reduced ellipsoid $\mathcal{E}_k = \mathcal{E} \cap \mathcal{S}_k$ to be \mathbf{x}_{k+1} .

Now we describe how to calculate \mathbf{x}_{k+1} in Algorithm 2. Notice that the linear manifold \mathcal{S}_k in (4.1) can be expressed by (4.5) where A_k and \mathbf{b}_k are still given by (4.7), (4.4) and (4.3). Also notice that the point \mathbf{a} in \mathcal{R}^n is corresponding to $\mathbf{a}_l = (\|\mathbf{v}_k\|, 0)^T$ in the \mathbf{l} space. Due to the invariance property of the projection under orthogonal transformations and the fact that $H_k^T H_k = I$, if the projection \mathbf{a}_l^* of \mathbf{a}_l onto the ellipsoid $\mathcal{E}_k^{(l)}$ in (4.6) is obtained, the \mathbf{x}_{k+1} in Algorithm 2 is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + H_k \mathbf{a}_l^*. \quad (5.8)$$

Therefore our calculation of \mathbf{x}_{k+1} in Algorithm 2 can be divided into two steps: the first step is to compute the projection \mathbf{a}_l^* of $\mathbf{a}_l = (\|\mathbf{v}_k\|, 0)^T$ onto $\mathcal{E}_k^{(l)}$ in (4.6) and the second step is to calculate \mathbf{x}_{k+1} from \mathbf{a}_l^* .

At the first step, to compute \mathbf{a}_l^* , we assume that the eigendecomposition of the 2×2 matrix A_k in (4.7) is

$$A_k = Q^T D Q, \quad \text{where } D \text{ is diagonal and } Q^T Q = I. \quad (5.9)$$

Under the orthogonal transformation $\mathbf{l} \rightarrow \mathbf{h} = Q\mathbf{l} + D^{-1}Q\mathbf{b}$, the ellipsoid $\mathcal{E}_k^{(l)}$ can be expressed by the form (5.1) with $\beta = (Q\mathbf{b}_k)^T D^{-1}(Q\mathbf{b}_k)$. The point \mathbf{a}_l in the \mathbf{l} space is corresponding to $\mathbf{a}_h = Q\mathbf{a}_l + D^{-1}Q\mathbf{b}_k$ in the \mathbf{h} space. Denote by \mathbf{a}_h^* the projection of \mathbf{a}_h onto $\mathcal{E}^{(h)}$, we also have that $\mathbf{a}_h^* = Q\mathbf{a}_l^* + D^{-1}Q\mathbf{b}_k$. Consequently, we have that

$$\mathbf{a}_l^* = Q^T(\mathbf{a}_h^* - D^{-1}Q\mathbf{b}_k). \quad (5.10)$$

By Lemma 1, the projection \mathbf{a}_h^* of \mathbf{a}_h onto $\mathcal{E}^{(h)}$ can be obtained via a quartic equation. Therefore after A_k and \mathbf{b}_k has been obtained, the computational work to obtain the projection \mathbf{a}_l^* of \mathbf{a}_l onto the 2-dimensional ellipsoid $\mathcal{E}_k^{(l)}$ is again negligible when n is relatively large.

At the second step, we may calculate \mathbf{x}_{k+1} from \mathbf{a}_l^* directly by (5.8). To avoid the explicit storage of H_k , however, we express \mathbf{x}_{k+1} by the form (2.5) and treat Algorithm 2 as a special case of The General Algorithm described in Section 3. Assume that $\mathbf{a}_l^* = (a_{l,1}^*, a_{l,2}^*)^T$. It follows from (5.8), the definition of H_k and (4.3) that

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_{l,1}^* \mathbf{p}_k + a_{l,2}^* \mathbf{q}_k = \mathbf{x}_k + \frac{a_{l,2}^*}{\|\mathbf{z}_k\|} \mathbf{u}_k + \left[\frac{a_{l,1}^*}{\|\mathbf{v}_k\|} - \frac{\mathbf{u}_k^T \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} \frac{a_{l,2}^*}{\|\mathbf{z}_k\|} \right] \mathbf{v}_k. \quad (5.11)$$

On the other hand, the definitions of \mathbf{w}_k , \mathbf{c}_k and \mathbf{v}_k in Section 2 indicate that

$$\mathbf{w}_k = \mathbf{c}_k - \mathbf{a} = \mathbf{x}_k - \gamma_k \mathbf{u}_k - \mathbf{a} = -\gamma_k \mathbf{u}_k - \mathbf{v}_k. \quad (5.12)$$

By (5.12) and the definition of \mathbf{v}_k , the \mathbf{x}_{k+1} in (2.5) can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k + \eta_k \mathbf{w}_k = \mathbf{x}_k - \eta_k \gamma_k \mathbf{u}_k + (1 - \eta_k) \mathbf{v}_k. \quad (5.13)$$

Comparing (5.11) and (5.13), we can then calculate the \mathbf{x}_{k+1} by (2.5) and (5.12) with

$$\eta_k = 1 - \frac{a_{l,1}^*}{\|\mathbf{v}_k\|} + \frac{\mathbf{u}_k^T \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} \frac{a_{l,2}^*}{\|\mathbf{z}_k\|} \quad (5.14)$$

and

$$\gamma_k = -\frac{a_{l,2}^*}{\|\mathbf{z}_k\|} \frac{1}{\eta_k}. \quad (5.15)$$

The above equivalent treatment also convenes us in designing a safeguard for Algorithm 2. If the γ_k in (5.15) is negative or tiny (this is sometimes the case in our numerical experiments though seldomly), we can turn to use (4.10) and carry out one step by the maximal 2-dimensional inside ball algorithm.

Using (4.4) and (4.3), the matrix A_k and \mathbf{b}_k in the expression of $\mathcal{E}_k^{(l)}$ can be also calculated without the explicit storage of H_k . If we do not consider the difference in the calculation of γ_k , the computation amount of Algorithm 2 and that of Algorithm 1 are identical since they require the same vector operations; namely, both of them require only 1 matrix-vector multiplication and 12 vector-vector operations or scalar-vector multiplications. Due to the minimal property, however, Algorithm 2 is expected to perform better than 1. When we use Algorithm 2 to solve the example in Section 3, a solution with the same precision is achieved at the 79-th iteration.

6. Hybrid Projection Algorithms

Algorithm 1 and Algorithm 2 avoid the direct estimate to the spectral radius $\rho(A)$ and are more efficient than the Lin-Han algorithm. Algorithm 2 seems to be optimal since the distance function $d(\mathbf{a}, \mathbf{x})$ achieves the maximal decrease in the 2-dimensional ellipsoid \mathcal{S}_k at every iteration. However, the following example shows that Algorithm 2 produces some kind of zigzags. Consider the 3-dimensional example with

Example 3. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^3 : x_1^2 + 100x_2^2 + 10000x_3^2 = 1.0101\}$, $\mathbf{a} = (2, 1.01, 1.0001)^T$. The initial point is $\mathbf{x}_0 = (0.01, 0.01, 0.01)^T$. The exact projection of \mathbf{a} onto \mathcal{E} is $\mathbf{x}^* = (1, 0.01, 0.0001)^T$.

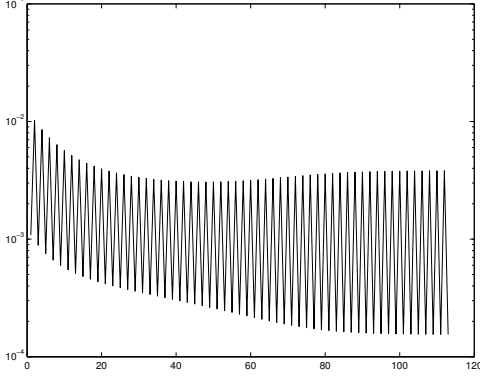


Figure 2. The $\{\gamma_k\}$ by Algorithm 2

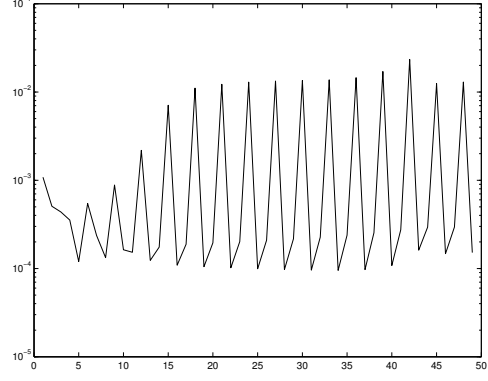


Figure 3. The $\{\gamma_k\}$ by Algorithm 3

Even for this 3-dimensional example, Algorithm 2 takes 113 iterations to reach the stopping condition (2.8) with $\varepsilon = 10^{-6}$. Denote again $\mathbf{e}_k = \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|}$ and the matrices

$$\bar{M}_k = [\mathbf{e}_{2k}, \mathbf{e}_{2k+2}, \mathbf{e}_{2k+4}], \quad \tilde{M}_k = [\mathbf{e}_{2k+1}, \mathbf{e}_{2k+3}, \mathbf{e}_{2k+5}].$$

We found that both the determinants of \bar{M}_k and \tilde{M}_k tend to zero as k increases. This shows that the iterations generalized by Algorithm 2 tend to two 2-dimensional reduced ellipsoids alternately. Meanwhile, the sequence $\{\gamma_k\}$ also tend to two different values, as shown in Figure 2. The same phenomenon is observed for Algorithm 2 in Example 1 in Section 3.

Instead of establishing strict theoretical results for the above observations, we are interested in this paper to find more efficient algorithms. A naive idea to avoid the zigzagging phenomenon of Algorithm 2 is to use one iteration of Algorithm 1 after every two iterations of Algorithm 2. We then obtain the following simple hybrid projection algorithm, where for some given \mathbf{x}_k , $\gamma_k^{(1)}$ and $\gamma_k^{(2)}$ stand for the values of γ_k given by Algorithm 1 and Algorithm 2, respectively.

Algorithm 3 (simple hybrid projection algorithm)

At step 2 in The General Algorithm, set $\gamma_{3k+i} = \gamma_{3k+i}^{(2)}$ for $i = 0, 1$ and $\gamma_{3k+2} = \gamma_{3k+2}^{(1)}$.

Although its basic idea is simple, our numerical experiments on a collection of test problems showed that Algorithm 3 is much more efficient than Algorithm 1 and Algorithm 2. For instance, to solve Example 3 with the same precision, Algorithm 3 only requires 49 iterations, which is significantly smaller than the number by Algorithm 2. To solve Example 1 in Section 3, Algorithm

3 only takes 27 iterations. At the same time, we notice that the value of γ_k changes frequently in Algorithm 3 (see Figure 3 for $\{\gamma_k\}$ in Example 3).

A possible explanation for the success of Algorithm 3 is that, the inexactness in the projection of \mathbf{a} on the 2-dimensional reduced ellipsoid \mathcal{E}_{3k+2} introduced by Algorithm 1 can make \mathcal{E}_{3k+3} and \mathcal{E}_{3k+4} be much different from \mathcal{E}_{3k} and \mathcal{E}_{3k+1} and hence help Algorithm 2 continuously achieve big decreases in the distance function $d(\mathbf{a}, \mathbf{x}_k)$. It is interesting to note that a similar idea has been used in the steepest descent method and leads to significant numerical improvement (see [3]).

The degree of inexactness in Algorithm 3 depends on the values of $\gamma_{3k+2}^{(1)}$ and $\gamma_{3k+2}^{(2)}$. If $\gamma_{3k+2}^{(1)} \approx \gamma_{3k+2}^{(2)}$, Algorithm 3 fails to bring enough inexactness. On the other hand, we see that there are many other ways to control the inexactness (for example, to multiply $\gamma_k^{(2)}$ by some positive constant less than 1). In addition, from Figure 3 we have some worry that the sequences $\{\gamma_k\}$ and $\{\mathbf{e}_k\}$ in Algorithm 3 may also sink into some type of cycle. Therefore we propose the following general hybrid projection algorithm.

Algorithm 4 (general hybrid projection algorithm)

At step 2 of The General Algorithm, compute γ_k by some positive function $\psi(\gamma_k^{(1)}, \gamma_k^{(2)})$ of $\gamma_k^{(1)}$ and $\gamma_k^{(2)}$.

The above algorithm includes Algorithm 1, Algorithm 2 and Algorithm 3 as its members. Now we discuss how to choose the function ψ . To guarantee the existence of \mathbf{x}_{k+1} and $d(\mathbf{a}, \mathbf{x}_{k+1}) < d(\mathbf{a}, \mathbf{x}_k)$, we impose the following condition

$$\gamma_k \leq \max(\gamma_k^{(1)}, \gamma_k^{(2)}). \quad (6.1)$$

If this relation holds, it is easy to know by continuity that the line segment connecting \mathbf{a} and $\mathbf{c}_k = \mathbf{x}_k - \gamma_k \mathbf{u}_k$ must have an intersection point with $\Omega(\mathcal{E}_k)$. By (6.1) and the positivity of ψ , we can express γ_k as

$$\gamma_k = c_k^{(1)} \gamma_k^{(1)} + c_k^{(2)} \gamma_k^{(2)}, \quad (6.2)$$

where $c_k^{(1)}$ and $c_k^{(2)}$ are such that

$$c_k^{(1)} \geq 0, \quad c_k^{(2)} \geq 0, \quad c_k^{(1)} + c_k^{(2)} \leq 1. \quad (6.3)$$

To ensure the convergence of the algorithm, we require that

$$c_k^{(1)} + c_k^{(2)} \geq \tau, \quad \text{for some } \tau \in (0, 1] \text{ and all } k. \quad (6.4)$$

Under these requirements on $c_k^{(1)}$ and $c_k^{(2)}$, we will show in the next section that the algorithm is globally convergent and the convergence is linear.

In this paper, we are particularly interested in the following four-parameter family of hybrid projection algorithms:

$$\gamma_k = \begin{cases} \gamma_k^{(2)}, & \text{if } \text{mod}(k, m_1 + m_2) < m_1; \\ c_1 \gamma_k^{(1)} + c_2 \gamma_k^{(2)}, & \text{otherwise,} \end{cases} \quad (6.5)$$

where $m_1 \geq 1$ and $m_2 \geq 1$ are integers, and c_1 and c_2 are non-negative constants satisfying $0 < c_1 + c_2 \leq 1$. The formula (6.5) indicates that the algorithm will carry out m_2 inexact 2-dimensional projection steps after every m_1 steps of Algorithm 2. In Section 8, we will find that some methods in the family (6.5) are more efficient than the simple hybrid projection algorithm. Here we would like to note that Algorithm (6.5) with the choice (8.2) only require 23 iterations for Example 1.

7. Linear Convergence

Lin and Han [9] proved the global convergence for their algorithm under the condition (3.2) on γ_k . In the following we will establish the linear convergence of the general hybrid projection algorithm with γ_k given by (6.2) under the assumptions (6.3) and (6.4) on $c_1^{(k)}$ and $c_2^{(k)}$. Consequently, the Lin-Han Algorithm and Algorithms 1-3 are all linearly convergent.

For any nonzero vectors \mathbf{x} and \mathbf{y} in \mathcal{R}^n , define the angle between \mathbf{x} and \mathbf{y} as

$$\theta(\mathbf{x}, \mathbf{y}) = \arccos\left(\frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}\right), \quad 0 \leq \theta(\mathbf{x}, \mathbf{y}) \leq \pi. \quad (7.1)$$

For any $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$, we denote the angles

$$\nu_k = \theta(\mathbf{a} - \mathbf{x}_k, A\mathbf{x}_k + \mathbf{b}), \quad \theta_k = \theta(\mathbf{a} - \mathbf{x}_k, \mathbf{x}^* - \mathbf{a}), \quad (7.2)$$

where \mathbf{x}^* is the projection of \mathbf{a} on \mathcal{E} as before. In the following, Lemmas 2 and 3 aim to provide a lower bound for the decrease $d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})$ by the angel ν_k . Lemma 5, that calls Lemma 4, estimates the upper bound for the distance $d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)$ by the angel θ_k . Then using the relation $\nu_k \geq \theta_k$, as shown in Lemma 6, we establish the linear convergence of the general hybrid projection algorithm in Theorem 7.

Denote by κ and $\lambda_{\min}(A)$ the condition number and the minimal eigenvalue of A , respectively. Define

$$\bar{\alpha} = \alpha + \mathbf{b}^T A^{-1} \mathbf{b}. \quad (7.3)$$

Under the transformation $\mathbf{x} \rightarrow \mathbf{y} = \mathbf{x} + A^{-1} \mathbf{b}$, we can express $\Omega(\mathcal{E})$ as

$$\bar{\Omega} = \{\mathbf{y} \in \mathcal{R}^n : \mathbf{y}^T A \mathbf{y} = \bar{\alpha}\}. \quad (7.4)$$

Then we can obtain

$$\begin{aligned} \min_{\mathbf{x} \in \Omega(\mathcal{E})} \|A\mathbf{x} + \mathbf{b}\| &= \min_{\mathbf{y} \in \bar{\Omega}} \|A\mathbf{y}\| = \min_{\mathbf{y} \in \bar{\Omega}} \sqrt{(A^{\frac{1}{2}}\mathbf{y})^T A (A^{\frac{1}{2}}\mathbf{y})} \\ &= \min_{\mathbf{y} \in \bar{\Omega}} \sqrt{\lambda_{\min}(A) \|A^{\frac{1}{2}}\mathbf{y}\|^2} = \sqrt{\lambda_{\min}(A)} \bar{\alpha}. \end{aligned} \quad (7.5)$$

Lemma 2 *For Algorithm 1, there exists some positive constant c_3 such that*

$$d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}) \geq c_3 \sin^2 \frac{\nu_k}{2}, \quad \text{for all } k, \quad (7.6)$$

where

$$c_3 = \frac{2 d(\mathbf{a}, \mathbf{x}^*)}{1 + c_4 d(\mathbf{a}, \mathbf{x}^*)} \quad \text{and} \quad c_4 = (\rho(A)/\bar{\alpha})^{\frac{1}{2}} \kappa^{\frac{1}{2}}. \quad (7.7)$$

Proof Denote by \mathbf{x}_s the intersection of the line segment $\mathcal{L}(\mathbf{a}, \mathbf{c}_k)$ and the boundary of the 2-dimensional ball $\mathcal{B}_2(\gamma_k)$ in (4.2) (see Figure 4). Then we have that $\|\mathbf{x}_s - \mathbf{c}_k\| = \|\mathbf{x}_k - \mathbf{c}_k\| = \gamma_k \|\mathbf{u}_k\|$. Noting that $\mathcal{B}_2(\gamma_k) \subset \mathcal{E}_k$ and considering the triangle formed by the points \mathbf{a} , \mathbf{c}_k and \mathbf{x}_k , we can get that

$$\begin{aligned} d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}) &= [d(\mathbf{a}, \mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|] - [d(\mathbf{a}, \mathbf{x}_{k+1}) + \|\mathbf{x}_s - \mathbf{c}_k\|] \\ &\geq [d(\mathbf{a}, \mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|] - \|\mathbf{a} - \mathbf{c}_k\| \\ &\geq \frac{[d(\mathbf{a}, \mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|]^2 - \|\mathbf{a} - \mathbf{c}_k\|^2}{2 [d(\mathbf{a}, \mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|]} \\ &= \frac{d(\mathbf{a}, \mathbf{x}_k) \|\mathbf{x}_k - \mathbf{c}_k\| [1 - \cos(\pi - \nu_k)]}{d(\mathbf{a}, \mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|} \\ &= \frac{2 \sin^2 \frac{\nu_k}{2}}{[d(\mathbf{a}, \mathbf{x}_k)]^{-1} + [\gamma_k \|\mathbf{u}_k\|]^{-1}}. \end{aligned} \quad (7.8)$$

From the above relation, (4.10), (4.11), the definition of \mathbf{u}_k and (7.5), we know that (7.6) holds. *q.e.d.*

Lemma 3 *Consider Algorithm 4 with γ_k given by (6.2). If $c_1^{(k)}$ and $c_2^{(k)}$ satisfy (6.3) and (6.4), we have that*

$$d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}) \geq c_3 \tau \sin^2 \frac{\nu_k}{2}, \quad \text{for all } k. \quad (7.9)$$

Proof For any fixed \mathbf{x}_k , denote by $\mathbf{x}_{k+1}^{(1)}$ and $\mathbf{x}_{k+1}^{(2)}$ the points generated by Algorithm 1 and Algorithm 2, respectively. Noting that $d(\mathbf{a}, \mathbf{x}_{k+1}^{(2)}) = \min\{d(\mathbf{a}, \mathbf{x}) : \mathbf{x} \in \mathcal{E}_k\} \leq d(\mathbf{a}, \mathbf{x}_{k+1}^{(1)})$, we have by this and Lemma 2 that

$$d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i)}) \geq c_3 \sin^2 \frac{\nu_k}{2}, \quad \text{for } i = 1, 2 \text{ and all } k. \quad (7.10)$$

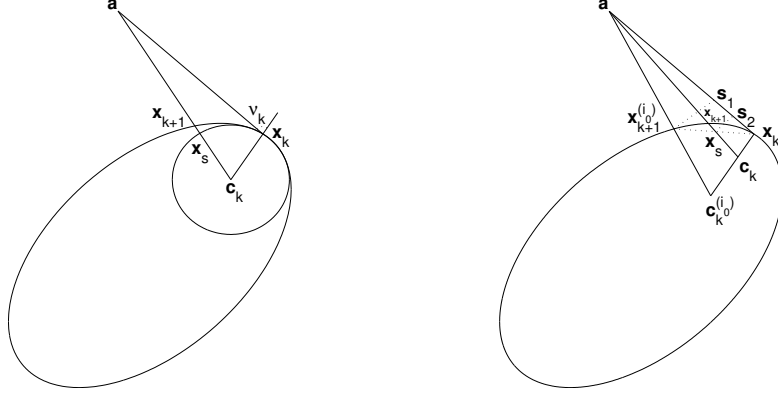


Figure 4. Diagram of proof of Lemma 2 Figure 5. Diagram of proof of Lemma 3

The relations (6.2) and (6.3) imply that $\gamma_k \leq \max(\gamma_k^{(1)}, \gamma_k^{(2)})$. Define $\mathbf{x}_{k+1}(\gamma) = \mathcal{L}(\mathbf{a}, \mathbf{x}_k - \gamma \mathbf{u}_k) \cap \Omega(\mathcal{E}_k)$. Since $\gamma_k^{(2)}$ is the unique minimizer of the function $d(\mathbf{a}, \mathbf{x}_{k+1}(\gamma))$ such that $\mathbf{x}_{k+1}(\gamma) \subset \Omega(\mathcal{E}_k)$, we know that $d(\mathbf{a}, \mathbf{x}_{k+1}(\gamma))$ is monotonically decreasing as γ moves from $\gamma_k^{(1)}$ to $\gamma_k^{(2)}$. Consequently, if

$$\gamma_k \geq \min(\gamma_k^{(1)}, \gamma_k^{(2)}), \quad (7.11)$$

we have that $d(\mathbf{a}, \mathbf{x}_{k+1}) \leq d(\mathbf{a}, \mathbf{x}_{k+1}^{(1)})$ and hence (7.9) is true.

If (7.11) does not hold, we have by (6.2) and (6.3) that

$$\tau \gamma_k^{(i_0)} \leq \gamma_k \leq \gamma_k^{(i_0)}, \quad (7.12)$$

where $i_0 \in \{0, 1\}$ is such that $\gamma_k^{(i_0)} = \min(\gamma_k^{(1)}, \gamma_k^{(2)})$. Equivalently, we have that

$$\tau \|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\| \leq \|\mathbf{c}_k - \mathbf{x}_k\| \leq \|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|, \quad (7.13)$$

where $\mathbf{c}_k^{(i_0)} = \mathbf{x}_k - \gamma_k^{(i_0)} \mathbf{u}_k$. Denote again by \mathbf{x}_s the intersection of $\mathcal{L}(\mathbf{a}, \mathbf{c}_k)$ and the line segment connecting \mathbf{x}_k and $\mathbf{x}_{k+1}^{(i_0)}$ (see Figure 5). Due to the convexity of $\Omega(\mathcal{E}_k)$, we know that \mathbf{x}_s belongs to the interior of \mathcal{E}_k and hence

$$d(\mathbf{a}, \mathbf{x}_{k+1}) < d(\mathbf{a}, \mathbf{x}_s). \quad (7.14)$$

For convenience, for any given vectors \mathbf{z}_1 , \mathbf{z}_2 , and \mathbf{z}_3 , we denote by $\angle \mathbf{z}_1 \mathbf{z}_2 \mathbf{z}_3$ the angle between $\mathbf{z}_1 - \mathbf{z}_2$ and $\mathbf{z}_3 - \mathbf{z}_2$. Note that $\angle \mathbf{a} \mathbf{c}_k \mathbf{x}_k > \angle \mathbf{a} \mathbf{c}_k^{(i_0)} \mathbf{x}_k$. If introducing a supplementary point $\mathbf{s} \subset \mathcal{L}(\mathbf{x}_k, \mathbf{x}_{k+1}^{(i_0)})$ such that $\mathcal{L}(\mathbf{s}, \mathbf{c}_k)$ is parallel to $\mathcal{L}(\mathbf{x}_{k+1}^{(i_0)}, \mathbf{c}_k^{(i_0)})$, we can see that

$$\frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|} > \frac{\|\mathbf{c}_k - \mathbf{x}_k\|}{\|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|}. \quad (7.15)$$

Now we introduce a supplementary point $\mathbf{s}_1 \in \mathcal{L}(\mathbf{a}, \mathbf{x}_k)$ such that $\|\mathbf{a} - \mathbf{s}_1\| = \|\mathbf{a} - \mathbf{x}_{k+1}^{(i_0)}\|$. Then we have that

$$\angle \mathbf{a} \mathbf{x}_{k+1}^{(i_0)} \mathbf{s}_1 = \angle \mathbf{a} \mathbf{s}_1 \mathbf{x}_{k+1}^{(i_0)} = \angle \mathbf{s}_1 \mathbf{x}_{k+1}^{(i_0)} \mathbf{x}_k + \angle \mathbf{a} \mathbf{x}_k \mathbf{x}_{k+1}^{(i_0)}, \quad (7.16)$$

$$\angle \mathbf{a} \mathbf{x}_{k+1}^{(i_0)} \mathbf{s}_1 + \angle \mathbf{a} \mathbf{s}_1 \mathbf{x}_{k+1}^{(i_0)} + \angle \mathbf{x}_k \mathbf{a} \mathbf{x}_{k+1}^{(i_0)} = \pi. \quad (7.17)$$

Substituting (7.16) into (7.17), we get that

$$2\angle \mathbf{s}_1 \mathbf{x}_{k+1}^{(i_0)} \mathbf{x}_k = (\pi - 2\angle \mathbf{a} \mathbf{x}_k \mathbf{x}_{k+1}^{(i_0)}) - \angle \mathbf{x}_k \mathbf{a} \mathbf{x}_{k+1}^{(i_0)}. \quad (7.18)$$

Similarly, if we introduce another supplementary point $\mathbf{s}_2 \in \mathcal{L}(\mathbf{a}, \mathbf{x}_k)$ such that $\|\mathbf{a} - \mathbf{s}_2\| = \|\mathbf{a} - \mathbf{x}_s\|$, we have that

$$2\angle \mathbf{s}_2 \mathbf{x}_{k+1}^{(i_0)} \mathbf{x}_k = (\pi - 2\angle \mathbf{a} \mathbf{x}_k \mathbf{x}_{k+1}^{(i_0)}) - \angle \mathbf{x}_k \mathbf{a} \mathbf{x}_s. \quad (7.19)$$

The relations (7.18), (7.19) and $\angle \mathbf{x}_k \mathbf{a} \mathbf{x}_s < \angle \mathbf{x}_k \mathbf{a} \mathbf{x}_{k+1}^{(i_0)}$ imply that $\angle \mathbf{s}_2 \mathbf{x}_{k+1}^{(i_0)} \mathbf{x}_k > \angle \mathbf{s}_1 \mathbf{x}_{k+1}^{(i_0)} \mathbf{x}_k$. Similarly to (7.15), we can prove that

$$\frac{\|\mathbf{s}_2 - \mathbf{x}_k\|}{\|\mathbf{s}_1 - \mathbf{x}_k\|} > \frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|}. \quad (7.20)$$

Therefore by (7.14), (7.20), (7.15) and (7.12), we obtain

$$\begin{aligned} \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i_0)})} &> \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_s)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i_0)})} = \frac{\|\mathbf{s}_2 - \mathbf{x}_k\|}{\|\mathbf{s}_1 - \mathbf{x}_k\|} \\ &> \frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|} > \frac{\|\mathbf{c}_k - \mathbf{x}_k\|}{\|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|} \geq \tau, \end{aligned} \quad (7.21)$$

which, with (7.10), indicates the truth of (7.9). *q.e.d.*

To estimate the distance between $d(\mathbf{a}, \mathbf{x}_k)$ and $d(\mathbf{a}, \mathbf{x}^*)$, we require the following lemma.

Lemma 4 *Consider the n -dimensional ellipsoid \mathcal{E} in (1.2). For any $\mathbf{x}, \mathbf{y} \in \Omega(\mathcal{E})$ with $\mathbf{x} \neq \mathbf{y}$, we have that*

$$\cos \theta(\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{x} - \mathbf{y}) \leq \frac{1}{2}c_4\|\mathbf{x} - \mathbf{y}\|, \quad (7.22)$$

where c_4 is given in (7.7).

Proof Without loss of generality, we assume that $n = 2$ for otherwise consider the reduced ellipsoid of \mathcal{E} restricted to the 2-dimensional linear manifold $\{\mathbf{x} + (A\mathbf{x} + \mathbf{b}, \mathbf{x} - \mathbf{y}) \mathbf{r} : \mathbf{r} \in \mathcal{R}^2\}$. Further, by making the transformation $\mathbf{x} \rightarrow \mathbf{x} + A^{-1}\mathbf{b}$ and some orthogonal transformation, we assume that

$$\mathcal{E} = \{\mathbf{x} \in R^2 : \mathbf{x}^T A \mathbf{x} \leq \bar{\alpha}\}, \text{ where } A = \text{diag}(\beta^2, \delta^2) \text{ with } 0 < \beta \leq \delta, \quad (7.23)$$

and $\bar{\alpha}$ is still given in (7.3). Let $\varrho = \bar{\alpha}^{\frac{1}{2}}\beta^{-1}\delta^{-1}$. Then we can express any $\mathbf{x}, \mathbf{y} \in \Omega(\mathcal{E})$ as

$$\mathbf{x} = \varrho \begin{pmatrix} \delta \cos \alpha_1 \\ \beta \sin \alpha_1 \end{pmatrix}, \quad \mathbf{y} = \varrho \begin{pmatrix} \delta \cos \alpha_2 \\ \beta \sin \alpha_2 \end{pmatrix}.$$

Denote $\alpha_3 = \frac{\alpha_1 + \alpha_2}{2}$ and $\alpha_4 = \frac{\alpha_1 - \alpha_2}{2}$, we have by direct calculations that

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\|^2 &= \varrho^2 [\delta^2 (\cos \alpha_1 - \cos \alpha_2)^2 + \beta^2 (\sin \alpha_1 - \sin \alpha_2)^2] \\ &= 4\varrho^2 \sin^2 \alpha_4 (\delta^2 \cos^2 \alpha_3 + \beta^2 \sin^2 \alpha_3) \end{aligned} \quad (7.24)$$

$$\geq 4\varrho^2 \beta^2 \sin^2 \alpha_4 \quad (7.25)$$

and

$$\begin{aligned} (\mathbf{x} - \mathbf{y})^T A \mathbf{x} &= \varrho^2 \beta^2 \delta^2 [\cos \alpha_1 (\cos \alpha_1 - \cos \alpha_2) + \sin \alpha_1 (\sin \alpha_1 - \sin \alpha_2)] \\ &= \varrho^2 \beta^2 \delta^2 [1 - (\cos \alpha_1 \cos \alpha_2 + \sin \alpha_1 \sin \alpha_2)] \\ &= \varrho^2 \beta^2 \delta^2 [1 - \cos(\alpha_1 + \alpha_2)] \end{aligned} \quad (7.26)$$

$$= 2\varrho^2 \beta^2 \delta^2 \sin^2 \alpha_4. \quad (7.27)$$

In addition, we can get that

$$\|A\mathbf{x}\| = \varrho\beta\delta\sqrt{\beta^2 \cos^2 \alpha_1 + \delta^2 \sin^2 \alpha_1} \geq \varrho\beta^2\delta. \quad (7.28)$$

Thus by (7.1), $\mathbf{b} = \mathbf{0}$, (7.25)–(7.28) and the definition of ϱ , we obtain

$$\cos \theta(A\mathbf{x} + \mathbf{b}, \mathbf{x} - \mathbf{y}) = \frac{(\mathbf{x} - \mathbf{y})^T A \mathbf{x}}{\|\mathbf{x} - \mathbf{y}\|^2 \|A\mathbf{x}\|} \|\mathbf{x} - \mathbf{y}\| \leq \frac{\delta}{2\varrho\beta^2} \|\mathbf{x} - \mathbf{y}\| \leq \frac{\delta^2}{2\bar{\alpha}^{\frac{1}{2}}\beta} \|\mathbf{x} - \mathbf{y}\|.$$

If $n = 2$, we have that $\delta = \sqrt{\rho(A)}$ and $\beta = \sqrt{\lambda_{\min}(A)}$. If $n \geq 3$, in which case a 2-dimensional reduced ellipsoid is considered, we have similarly to (7.16) that $\delta \leq \sqrt{\rho(A)}$ and $\beta \geq \sqrt{\lambda_{\min}(A)}$. Consequently, (7.22) is always true. *q.e.d.*

With the help of Lemma 3, we can now estimate the distance between $d(\mathbf{a}, \mathbf{x}_k)$ and $d(\mathbf{a}, \mathbf{x}^*)$ by the angle θ_k in (7.2).

Lemma 5 Denote $\mathcal{S}(\mathbf{a}, \mathcal{E}) = \{\mathbf{x} \in \Omega(\mathcal{E}) : \theta(\mathbf{a} - \mathbf{x}, A\mathbf{x} + \mathbf{b}) \leq \frac{\pi}{2}\}$. If $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$, there exists some positive constant c_5 such that

$$d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*) \leq c_5 \sin^2 \frac{\theta_k}{2}. \quad (7.29)$$

Proof Define

$$\phi(\mathbf{x}) = \begin{cases} \theta(\mathbf{a} - \mathbf{x}^*, \mathbf{x} - \mathbf{x}^*), & \text{if } \mathbf{x} \neq \mathbf{x}^* \\ \frac{\pi}{2}, & \text{if } \mathbf{x} = \mathbf{x}^*. \end{cases}$$

It is easy to see that $\phi^* := \max\{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{S}(\mathbf{a}, \mathcal{E})\} \geq \frac{\pi}{2}$ since $\mathbf{x}^* \in \mathcal{S}(\mathbf{a}, \mathcal{E})$. In addition, notice that the point $\bar{\mathbf{x}}$ in \mathcal{E} satisfying $\theta(\mathbf{a} - \mathbf{x}^*, \bar{\mathbf{x}} - \mathbf{x}^*) = \pi$ does not belong to $\mathcal{S}(\mathbf{a}, \mathcal{E})$. Then we have by the compactness of $\mathcal{S}(\mathbf{a}, \mathcal{E})$ that $\phi^* < \pi$. Further, denote $\sigma_k = \theta(\mathbf{a} - \mathbf{x}^*, \mathbf{x}_k - \mathbf{x}^*)$ (see Figure 6). In a similar way, we can show that $\frac{\pi}{2} \geq \pi - (\theta_k + \sigma_k) \geq \xi^*$. It follows that for any $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$,

$$\sin \sigma_k \geq \sin \phi^* \quad \text{and} \quad \sin(\theta_k + \sigma_k) \geq \sin \xi^*. \quad (7.30)$$

From the triangle formed by \mathbf{a} , \mathbf{x}^* and \mathbf{x}_k , we have that

$$\frac{\|\mathbf{x}^* - \mathbf{x}_k\|}{\sin \theta_k} = \frac{d(\mathbf{a}, \mathbf{x}_k)}{\sin \sigma_k} = \frac{d(\mathbf{a}, \mathbf{x}^*)}{\sin(\theta_k + \sigma_k)}. \quad (7.31)$$

Noting that $\mathbf{a} - \mathbf{x}^*$ is parallel to $A\mathbf{x}^* + \mathbf{b}$, we have by Lemma 4 and the fact that $\mathbf{a} - \mathbf{x}^*$ is parallel to $A\mathbf{x}^* + \mathbf{b}$ that

$$-\cos \sigma_k = \cos(\pi - \sigma_k) \leq c_4 \|\mathbf{x}^* - \mathbf{x}_k\|. \quad (7.32)$$

Now, by (7.31), (7.32) and (7.30), we can obtain

$$\begin{aligned} \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)}{d(\mathbf{a}, \mathbf{x}^*)} &= \frac{\sin \sigma_k - \sin(\theta_k + \sigma_k)}{\sin(\theta_k + \sigma_k)} \\ &= \frac{\sin \sigma_k (1 - \cos \theta_k) - \cos \sigma_k \sin \theta_k}{\sin(\theta_k + \sigma_k)} \\ &\leq \frac{2 \sin \sigma_k \sin^2 \frac{\theta_k}{2} + \frac{1}{2} c_4 \|\mathbf{x}^* - \mathbf{x}_k\| \sin \theta_k}{\sin(\theta_k + \sigma_k)} \\ &= \frac{2 \sin^2 \sigma_k \sin^2 \frac{\theta_k}{2} + \frac{1}{2} c_4 d(\mathbf{a}, \mathbf{x}_k) \sin^2 \theta_k}{\sin(\theta_k + \sigma_k) \sin \sigma_k} \\ &= \frac{2 \sin^2 \sigma_k + 2 c_4 d(\mathbf{a}, \mathbf{x}_k) \cos^2 \frac{\theta_k}{2} \sin^2 \frac{\theta_k}{2}}{\sin(\theta_k + \sigma_k) \sin \sigma_k} \\ &\leq \frac{2(1 + c_4 d_{\max})}{\sin \xi^* \sin \phi^*} \sin^2 \frac{\theta_k}{2}. \end{aligned} \quad (7.33)$$

In the above, $d_{\max} = \max_{\mathbf{x} \in \mathcal{E}} d(\mathbf{a}, \mathbf{x}) < +\infty$. Therefore (7.29) holds with

$$c_5 = \frac{2(1 + c_4 d_{\max}) d(\mathbf{a}, \mathbf{x}^*)}{\sin \xi^* \sin \phi^*}, \quad (7.34)$$

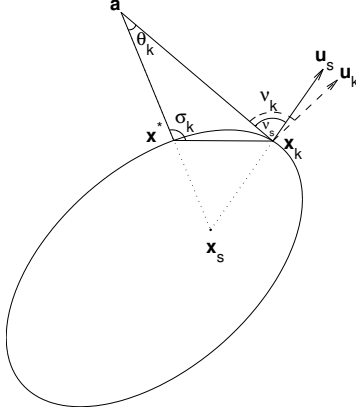


Figure 6. Diagram of proof of Lemmas 5 and 6

which completes the proof.

q.e.d.

To establish the linear convergence of the algorithm, we now need a relation between the angle ν_k and θ_k . Similarly to the proof of Theorem 3.8 in [9], we can show that $\nu_k \geq \theta_k$. In the following, we present a geometrical proof to this result (see also Figure 6 for the diagram of the proof).

Lemma 6 *For any $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$ with $\mathbf{x}_k \neq \mathbf{x}^*$, we have that $\nu_k \geq \theta_k$.*

Proof Denote $\xi_k = \theta(\mathbf{x}^* - \mathbf{x}_k, \mathbf{u}_k)$. Since \mathbf{u}_k is the normal direction of q at \mathbf{x}_k , we know that $\xi_k > \frac{\pi}{2}$. Denote by $\bar{\mathcal{S}}_k$ the 2-dimensional linear manifold including \mathbf{a} , \mathbf{x}^* and \mathbf{x}_k . Note that the direction \mathbf{u}_k does not necessarily lie in $\bar{\mathcal{S}}_k$. We then introduce a supplementary direction $\mathbf{u}_s \in \bar{\mathcal{S}}_k$ such that the angle $\theta(\mathbf{x}^* - \mathbf{x}_k, \mathbf{u}_s)$ has the same size as ξ_k . Meanwhile, we denote by $\bar{\mathcal{C}}_k$ the cone $\mathbf{x}_k \cup \{\mathbf{y} \neq \mathbf{x}_k : \theta(\mathbf{x}^* - \mathbf{x}_k, \mathbf{y} - \mathbf{x}_k) = \xi_k\}$. Then we can see that

$$\nu_s \doteq \theta(\mathbf{a} - \mathbf{x}_k, \mathbf{u}_s) = \min\{\theta(\mathbf{a} - \mathbf{x}_k, \mathbf{y} - \mathbf{x}_k) : \mathbf{y} \in \bar{\mathcal{C}}_k \setminus \{\mathbf{x}_k\}\} \leq \nu_k. \quad (7.35)$$

On the other hand, since \mathbf{x}^* is the projection of \mathbf{a} on the ellipsoid, we have that $\theta(\mathbf{a} - \mathbf{x}^*, \mathbf{x}_k - \mathbf{x}^*) > \frac{\pi}{2}$. Consequently, the straight line passing \mathbf{a} and \mathbf{x}^* and the one $\{\mathbf{x}_k + t\mathbf{u}_s : t \in \mathcal{R}^1\}$ must cross at some point, still say \mathbf{x}_s . From the triangle formed by \mathbf{a} , \mathbf{x}_s and \mathbf{x}_k , we can get that

$$\nu_s \geq \theta_k. \quad (7.36)$$

Combining (7.35) and (7.36), we know the truth of this lemma.

q.e.d.

Now we are able to give the main theorem.

Theorem 7 *Consider Algorithm 4 with γ_k given by (6.2). If $c_k^{(1)}$ and $c_k^{(2)}$ satisfy (6.3) and (6.4), there exists some positive constant $c_6 < 1$ such that*

$$\frac{d(\mathbf{a}, \mathbf{x}_{k+1}) - d(\mathbf{a}, \mathbf{x}^*)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} \leq 1 - c_6. \quad (7.37)$$

Proof By Lemmas 3, 5 and 6, we have that

$$\frac{d(\mathbf{a}, \mathbf{x}_{k+1}) - d(\mathbf{a}, \mathbf{x}^*)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} = 1 - \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} \leq 1 - \frac{c_3\tau}{c_5}. \quad (7.38)$$

Substituting the values of c_i 's, we know that (7.37) holds with

$$c_6 = \frac{c_3\tau}{c_5} = \frac{\tau \sin \xi^* \sin \sigma^*}{\left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}} \kappa^{\frac{1}{2}} d(\mathbf{a}, \mathbf{x}^*)\right] \left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}} \kappa^{\frac{1}{2}} d_{\max}\right]}. \quad (7.39)$$

The proof is then completed. *q.e.d.*

When $\mathbf{x}_k \rightarrow \mathbf{x}^*$, we have that $\sigma_k \rightarrow \frac{\pi}{2}$, $\theta_k + \sigma_k \rightarrow \frac{\pi}{2}$ and $d(\mathbf{a}, \mathbf{x}_k) \rightarrow d(\mathbf{a}, \mathbf{x}^*)$. Consequently, from the proof of Lemma 5, the linear convergence constant in (7.37) can be approximated by $1 - \bar{c}_6$ where

$$\bar{c}_6 = \frac{\tau}{\left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}} \kappa^{\frac{1}{2}} d(\mathbf{a}, \mathbf{x}^*)\right]^2}. \quad (7.40)$$

The relation (7.40) indicates that the convergence becomes slower when the condition number of A becomes larger or the point \mathbf{a} to be projected is farther from the ellipsoid.

8. Numerical Experiments

A 10-dimensional example has been used before to show the efficiency of new projection algorithms. Now we provide some numerical results for higher dimensional problems. To convene our observation, we assume that the matrix A is diagonal and its diagonal entries are given by $a_{ii} = 10^{\frac{i-1}{n-1}ncond}$ ($i = 1, \dots, n$) and $ncond$ controls the condition number of the matrix A . The vector \mathbf{b} is set to $\mathbf{0}$ in our tests although the algorithms can apply to the case of nonzero \mathbf{b} . In case of nonzero \mathbf{b} , we need to find a feasible initial points in \mathcal{E} . We set $c = 0$ and $\alpha = 1$ so that the ellipsoid \mathcal{E} lies in the unit ball at the origin. Equivalently, given n and $ncond$, the ellipsoid used in our test is

$$\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^n : \sum_{i=1}^n 10^{\frac{i-1}{n-1}ncond} x_i^2 = 1\}.$$

For the choice of \mathbf{a} , we first generate a point $\tilde{\mathbf{a}} = (\tilde{a}_i)$ by

$$\tilde{a}_i = a_{ii}^{-\omega}, \quad (8.1)$$

where $\omega \geq 0$ is some parameter. Then we ask $\mathbf{x}^* = (\tilde{\mathbf{a}}^T A \tilde{\mathbf{a}})^{-\frac{1}{2}} \tilde{\mathbf{a}}$ to be the projection of \mathbf{a} in the ellipsoid \mathcal{E} . It is easy to see that the larger ω is, the

more \mathbf{x}^* tends to an eigenvector of the matrix A corresponding to its small eigenvalue. For each \mathbf{x}^* , we choose different ϑ 's for \mathbf{a} such that $\|\mathbf{a}\| = \vartheta$. Given the size ϑ of \mathbf{a} , the point \mathbf{a} can be calculated by

$$\mathbf{a} = \mathbf{x}^* + \frac{\vartheta^2 - \|\mathbf{x}^*\|^2}{1 + \sqrt{1 + \|A\mathbf{x}^*\|^2} [\vartheta^2 - \|\mathbf{x}^*\|^2]} A\mathbf{x}^*.$$

The above strategy enables us not only to control both the size and the direction of \mathbf{a} (by the parameters ω and ϑ) but also to know its exact projection \mathbf{x}^* . To sum up, the construction of our test problems depends on the four parameters n , $ncond$, ω and ϑ . In our tests, we fix $n = 10^4$ and vary the other parameters:

$$ncond \in \{2, 3, 4, 5\}, \quad \omega \in \{0, 1/8, 1/4, 1/2\}, \quad \vartheta \in \{2, 5, 10, 50\}.$$

We tested the Lin-Han Algorithm and Algorithms 1-4 with the MATLAB language (version 6.5.0). For all cases, the initial point is set to $\mathbf{x}_0 = \vartheta^{-\frac{1}{2}}\mathbf{a}$. The stopping condition is (2.8) with $\varepsilon = 10^{-6}$. For the Lin-Han Algorithm, we set $\gamma_k = (\rho(A))^{-1} = 10^{-ncond}$. As analyzed in Section 3, this choice of γ_k favors the comparison of the Lin-Han Algorithm since any under estimation of this value may deteriorate the performance of the algorithm. The parameters in Algorithm 4 are chosen as follows:

$$m_1 = 1, \quad m_2 = 1, \quad c_1 = 0.1, \quad c_2 = 0.8. \quad (8.2)$$

Nevertheless, good numerical results are also obtained with $(m_1, m_2, c_1, c_2) = (1, 1, 0.05, 0.9)$, *etc.* Generally, if $m_1 = m_2 = 1$ are fixed, the suggested arranges for c_1 and c_2 are that

$$c_2 \in [0.7, 0.9], \quad c_1 \in [0, 0.95 - c_2].$$

The iteration numbers required by the algorithms for each case are taken down in Table 2, where LH and Ai stand for the Lin-Han algorithm and Algorithm i , respectively. Since the computation amount per iteration required by each algorithm is similar, the algorithmic performance can basically be evaluated by the required iteration numbers. From Table 2, we make the following comments:

Regarding influence of $ncond$, ω and ϑ . In general, we see that the problem becomes more difficult as $ncond$ and ω increase. In other words, when the ellipsoid becomes more *flat*, it is more difficult to project those points close to the flat part of the ellipsoid. The influence of ϑ , namely, the size of \mathbf{a} , is different. If fixing $ncond$ and ω , the increase of ϑ leads to more projection iterations in most of the cases. However, for quite many cases such as $ncond =$

$ncond = 2$							$ncond = 3$				
LH	A1	A2	A3	A4	ω	ϑ	LH	A1	A2	A3	A4
72	49	40	19	17	0	2	209	141	118	44	43
110	74	62	25	25	0	5	316	212	184	62	57
134	90	76	29	29	0	10	382	256	231	77	65
163	109	94	35	31	0	50	434	290	280	86	77
82	54	45	19	19	1/8	2	269	173	148	52	41
127	82	70	26	29	1/8	5	435	278	246	83	48
157	101	88	32	31	1/8	10	559	357	324	104	51
194	124	110	40	35	1/8	50	716	456	440	137	87
94	58	50	22	21	1/4	2	344	208	186	65	49
146	89	80	31	31	1/4	5	580	349	322	104	64
181	110	100	37	33	1/4	10	773	465	436	137	121
225	136	126	43	39	1/4	50	1076	647	630	193	97
121	63	64	37	27	1/2	2	579	293	304	136	62
187	96	100	52	35	1/2	5	994	501	534	223	115
228	117	124	62	37	1/2	10	1337	672	728	295	166
278	142	152	73	43	1/2	50	1899	953	1052	409	165
$ncond = 4$							$ncond = 5$				
LH	A1	A2	A3	A4	ω	ϑ	LH	A1	A2	A3	A4
479	321	282	92	75	0	2	809	541	529	158	77
621	415	396	125	85	0	5	697	466	529	137	82
623	416	429	125	88	0	10	580	388	440	116	70
524	350	377	104	74	0	50	506	338	364	104	69
734	469	414	128	70	1/8	2	1732	1104	1018	290	141
1136	724	672	200	97	1/8	5	2255	1437	1466	386	141
1362	868	854	239	101	1/8	10	2069	1318	1546	362	126
1298	828	932	236	105	1/8	50	1299	828	1008	236	94
1068	643	586	188	76	1/4	2	3020	1815	1684	509	151
1834	1102	1036	311	120	1/4	5	5015	3012	2920	833	175
2485	1493	1452	425	173	1/4	10	6430	3862	3972	1058	247
3261	1959	2122	557	197	1/4	50	5404	3245	4288	899	203
2357	1183	1246	535	171	1/2	2	8700	4357	4622	1858	429
4352	2181	2354	937	253	1/2	5	16715	8367	9074	3415	546
6352	3182	3488	1312	379	1/2	10	25653	12839	14178	5071	775
10769	5391	6128	2119	360	1/2	50	49765	24902	29210	9349	800

Table 2. Numerical comparisons of five projection algorithms

4, $\omega \in \{0, 1/8\}$ and $ncond = 5$, $\omega \in \{0, 1/8, 1/4\}$, the required iterations for $\vartheta = 50$ are generally less than those for $\vartheta = 10$. It seems to us that for each case of $ncond$ and ω , the problem becomes eventually more difficult as \mathbf{a} gets farther away from the ellipsoid and then eventually easier after \mathbf{a} exceeds some distance.

Regarding efficiency of five projection algorithms. It is evident that the Lin-Han is the worst and Algorithm 4 is the best. Further, we see that the gain achieved by Algorithm 4 is bigger as the problem becomes more difficult. In the most difficult case that $ncond = 5$, $\omega = 1/2$ and $\vartheta = 50$, the iteration number required by Algorithm 4 is only about one sixty-third of the number by the Lin-Han algorithm. In other words, Algorithm 4 is less influenced by the difficulty of the problem. Algorithm 3 is the second best among the five algorithms. Comparing Algorithm 1 and Algorithm 2, we see that in *easy* cases, Algorithm 2 is better than Algorithm 1, whereas Algorithm 1 requires fewer iterations than Algorithm 2 in *difficult* cases.

9. Discussion

In this paper we have proposed several new algorithms for projection on a general ellipsoid by considering the 2-dimensional reduced ellipsoid at each iteration. To avoid the direct estimation of the spectral radius $\rho(A)$ in the Lin-Han algorithm, we provided the maximal 2-dimensional inside ball algorithm (Algorithm 1) and the sequential 2-dimensional projection algorithm (Algorithm 2). However, we found that the solution procedure of Algorithm 1 tends to some 2-dimensional reduced ellipsoid. For Algorithm 2, the iterations tend to two 2-dimensional reduced ellipsoids alternately. Therefore we investigated the hybrid use of the two algorithms and proposed the simple hybrid projection algorithm (Algorithm 3) and the general hybrid projection algorithm (Algorithm 4). Our numerical experiments show that Algorithms 1-4, even Algorithm 4, are much faster than the Lin-Han algorithm even when the spectral radius $\rho(A)$ is exactly known. To further improve Algorithm 4, we feel that one possible approach is to impose some conjugacy on the 2-dimensional reduced ellipsoids $\{\mathcal{E}_k\}$.

One disadvantage of the algorithms of this paper is that they require a feasible point of the ellipsoid \mathcal{E} . Assume that a feasible point, \mathbf{x}_f say, has been found. Then one can choose the intersection of \mathcal{E} and the line segment $\mathcal{L}(\mathbf{a}, \mathbf{x}_f)$ as an initial point. If the right-hand-side term $\mathbf{b} = \mathbf{0}$, a feasible point can be easily found in \mathcal{E} . However, this is not always the case with nonzero \mathbf{b} . In this case, to find a feasible point of \mathcal{E} and use the algorithms, one may need to reduce the function $q(\mathbf{x})$ from some infeasible point with the help of some minimization algorithm. Therefore it may be interesting to find some infeasible projection algorithms in which a feasible point is not necessary.

It is obvious that the maximal 2-dimensional inside ball algorithm can be extended to the problem of calculating the distance between two ellipsoids considered in Lin and Han [10],

$$\begin{aligned} \min \quad & \|\mathbf{x} - \mathbf{y}\| \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{E}, \mathbf{y} \in \bar{\mathcal{E}}. \end{aligned}$$

Hence the similar disadvantage of estimating the spectral radius of some ellipsoid in their algorithm can be avoided. Some kind of extension of the sequential 2-dimensional ellipsoid projection algorithm to such problem is also possible. For example, assuming that $\mathbf{x}_k \in \mathcal{E}$ and $\mathbf{y}_k \in \bar{\mathcal{E}}$ has been obtained at the k -th iteration, we can construct a maximal 2-dimensional inside ball of \mathcal{E} at \mathbf{x}_k . Denote the center of this ball to be \mathbf{c}_k . Then we can take $\mathbf{y}_{k+1} \in \bar{\mathcal{E}}$ to be the projection of \mathbf{c}_k on $\bar{\mathcal{E}}$ and then let $\mathbf{x}_{k+1} = \Omega(\mathcal{E}) \cap \mathcal{L}(\mathbf{c}_k, \mathbf{y}_{k+1})$. Therefore faster algorithms for the above problem should also be able to be obtained. Nevertheless, it still remains to study how to design the most efficient algorithms. In addition, it may be also interesting to investigate how to extend the idea of the algorithms proposed in this paper to solve the projection problem on a general convex set. More recent work on this aspect can be seen in Lin [8] and Lin and Han [11].

Acknowledgements. The author is very grateful to Professor Roger Fletcher for many useful discussions with him and for his kindness while the author was visiting University of Dundee. Many thanks are also due to the two anonymous referees, whose comments and suggestions improve the quality of this paper greatly.

References

- [1] E. G. Birgin, J. M. Martínez, and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization*, 10 (2000), pp. 1196-1211.
- [2] Y. H. Dai and R. Fletcher, New Algorithms for Singly Linearly Constrained Quadratic Programs Subject to Lower and Upper Bounds, Numerical Analysis Report NA/216, Department of Mathematics, University of Dundee, 2003.
- [3] Y. H. Dai and Y. Yuan, Alternate minimization gradient method, *IMA Journal of Numerical Analysis*, 23 (2003), pp. 377-393.
- [4] R. L. Dykstra, An algorithm for restricted least-squares regression, *Journal of the American Statistical Association*, 78 (1983), pp. 837-842.

- [5] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1991.
- [6] S. J. Grotzinger and C. Witzgall, Projections onto order simplexes, *Applied Mathematics and Optimization*, 12 (1984), pp. 247-270.
- [7] S.-P. Han and G. Lou, A parallel algorithm for a class of convex programs, *SIAM Journal on Control and Optimization*, 26 (1988), pp. 345-355.
- [8] A. Lin, A class of methods for projection on a convex set, *Advanced Modeling and Optimization*, 5:3 (2003), pp. 211-221.
- [9] A. Lin and S.-P. Han, Projection on an ellipsoid, Research report, Department of Mathematical Sciences, The Johns Hopkins University, 2001.
- [10] A. Lin and S.-P. Han, On the distance between two ellipsoids, *SIAM Journal on Optimization*, 13 (2002), pp. 298-308.
- [11] A. Lin and S.-P. Han, A class of methods for projection on the intersection of several ellipsoids, *SIAM Journal on Optimization*, 15 (2005) pp. 129-138.
- [12] P. M. Pardalos and N. Kover, An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds, *Mathematical Programming*, 46 (1990), pp. 321-328.