Projection onto quadratic hypersurfaces

Loïc Van Hoorebeeck[a], P.-A. Absil[a], Anthony Papavasiliou[b]

[a]ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium.
[b]CORE, UCLouvain, Louvain-la-Neuve, Belgium.

October 16, 2021

Abstract

We address the problem of projecting a point onto a quadratic hypersurface, more specifically a central quadric. We show how this problem reduces to finding a given root of a scalar-valued nonlinear function. We completely characterize one of the optimal solutions of the projection as either the unique root of this nonlinear function on a given interval, or as a point that belongs to a finite set of computable solutions. We then leverage this projection and the recent advancements in splitting methods to compute the projection onto the intersection of a box and a quadratic hypersurface with alternating projections and Douglas-Rachford splitting methods. We test these methods on a practical problem from the power systems literature, and show that they outperform IPOPT and Gurobi in terms of objective, execution time and feasibility of the solution.

Keywords: quadric, quadratic surfaces, nonconvex projection, Douglas-Rachford splitting, alternating projections.

1 Introduction

This paper discusses the projection of a given point onto a nonsingular quadratic hypersurface, or nonsingular *quadric*. Quadrics are a natural generalization of hyperplanes. The projection onto a quadric appears, e.g., in splitting algorithms for the projection between a quadric—or the Cartesian product of quadrics—and a polytope. This problem has direct applications, *e.g.*, in power systems [29], where the power losses can be approximated as a quadratic hypersurface. Numerical experiments on such problems are developed in this paper. Other applications of quadratic projections emerge in the context of the security region of gas networks [26], or in local learning methods [9].

It is therefore intriguing that few studies of this problem can be found in the literature. Indeed, projections onto quadratic surfaces have been studied for the 2D and 3D cases, see *e.g.*, [23, 22, 18]. However, to the best of our knowledge, the extension to an arbitrary dimension has not been pursued, with the exception of the short discussion at the end of [22] and in [27]. Although the method proposed in [27] can handle the singular case, *i.e.*, the case where the matrix that defines the quadratic surface is singular, it does not always return the exact projection. Moreover, the two-level iterative scheme that is proposed in [27] can be computationally expensive.

The structure of this paper is twofold. Firstly, we tackle the problem of projecting onto an n-dimensional nonsingular quadric. Secondly, we leverage this projection in the context of splitting methods.

The projection considered in the first part of this paper (Section 2) is not unique in general, due to the nonconvexity of the feasible set. This implies that we cannot rely on first or second-order methods, since such methods may converge to a local minimum. This projection can also be dealt with by black-box (commercial) solvers, *e.g.*, Gurobi or IPOPT [17, 31], however these methods suffer from two main problems: i) the execution time rockets when the dimension of the problem increases to mid or large-scale sizes—this phenomenon is highlighted in our numerical results—and ii) Gurobi is not a local method, and does not exploit the local structure of the problem, even if in certain applications, the starting point is close to the feasible set. The first problem is highlighted by the power systems application that is considered here, where the projection step is only a small part of the overall procedure, which renders execution time an important factor in our analysis.

Using the Lagrange multiplier technique, we reduce this quadratically constrained quadratic program (QCQP) to the problem of finding the roots of a nonlinear real function. Then, we completely characterize the solutions of the nonconvex projection, and compute one of the solutions of this projection as either the (unique) root of a scalar function on a computable interval, or among a finite set of closed-form solutions. We also show how to find this root using Newton's method, which guarantees a quadratic convergence. Thus, the proposed method provides an efficient way to obtain the exact projection onto a nonsingular quadric. Finally, to further reduce execution time, we also introduce a heuristic based on a geometric construction. This allows us to quickly map a point to the quadric. We detail two variants of this heuristic.

We note that our proposed approach for projecting onto a quadric is not unusual. For example, [16, §6.2.1] uses a similar construction for the problem of least squares minimization over a sphere. However, this problem is easier to tackle than ours, since the (unique) solution of this convex problem is the (unique) root of the *secular equation* defined by the KKT conditions. In [11, §7.3], the authors also use a similar procedure, and taxonomy of secular equations, for finding the ℓ_2-norm model minimizer. However, while the discussion is analogous to what is proposed in this paper, *i.e.*, searching for a specific root of a given scalar-valued nonlinear function on a specific domain, the domain and the function are different in our work. Moreover, our discussion on degenerate cases is not present in [11], since such cases do not appear in the problem that the authors tackle, which is linked to the trust-region subproblem.

In the second part of this paper (Section 3) we test our method of projecting onto a quadric in order to solve the problem of projecting onto the intersection between a polytope and a quadric. Both projections can be easily computed: the first one is even trivial if we consider a box, and the second one is efficiently obtained with the method proposed in the first part of the paper. We then leverage the rich literature on splitting methods for nonconvex programming. We consider, in this work, two splitting methods: the alternating projections and Douglas-Rachford splitting. References for these schemes can be found in [4, 5, 13, 20, 19], and in [5, 21], respectively.

Depending on the splitting method considered, and whether or not we use exact projection onto the quadric or a heuristic, we detail five different methods for projecting a point onto the intersection of a box and a quadric. We analyse these methods in Section 4. The five methods are benchmarked against IPOPT in the ellipsoidal and hyperboloidal case, both for small and large-scale problems. In these experiments, we observe that one of the proposed methods, namely the alternating projections with exact projections, attains the best objective. We also observe that the alternating projections used with one of the heuristics (the gradient-based heuristic) reaches competitive objectives in a reduced amount of run time. All the methods considered outperform IPOPT in terms of execution time, with a difference of several orders of magnitude. Finally, we benchmark one of the proposed methods against Gurobi. We use Gurobi in order to find the optimal solution for a problem inspired from the power systems literature. Since, in this context, the starting point is close to the feasible set, our proposed method clearly outperforms Gurobi, both in terms of execution time and objective. Using the lower bound computed by Gurobi, we can also conclude that, in the context of this specific problem, the proposed method finds the optimal solution, even if there is no guarantee for finding it in general.

2 Projection onto a quadric

2.1 Problem formulation

In this section, the problem of interest is introduced. This problem consists in the projection of a given point $\tilde{\boldsymbol{x}}^0 \in \mathbb{R}^n$ onto a feasible set $\mathcal{Q}$:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \; & \left|\left|\boldsymbol{x} - \tilde{\boldsymbol{x}}^0\right|\right|_2^2 \\ \text{s.t. } \; & \boldsymbol{x} \in \mathcal{Q}, \end{aligned} \tag{1}$$

where $\mathcal{Q}$ is a nonempty and non-cylindrical central quadric [25, Theorem 3.1.1]. In other words, $\mathcal{Q}$ is nonempty and there exists a quadratic function

$$\Psi : \mathbb{R}^n \to \mathbb{R} : \boldsymbol{x} \mapsto \Psi(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{B} \boldsymbol{x} + \boldsymbol{b}^\intercal \boldsymbol{x} + c,$$

with $\boldsymbol{B}$ nonsingular and $c \neq \frac{\boldsymbol{b}^\intercal \boldsymbol{B}^{-1} \boldsymbol{b}}{4}$, such that

$$\mathcal{Q} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \Psi(\boldsymbol{x}) = 0\} = \Psi^{-1}(0). \tag{2}$$

See [25, §3.1] and [3, Chapter 21] for a complete classification of quadrics.

This quadratic surface, or *quadric*, is denoted as quadric with middle point, in the sense of [3]. The middle point or *centre*, $\boldsymbol{d}$, corresponding to the centre of symmetry, is computed as $-\frac{\boldsymbol{B}^{-1}\boldsymbol{b}}{2}$, and the condition $c \neq \frac{\boldsymbol{b}^\intercal \boldsymbol{B}^{-1}\boldsymbol{b}}{4}$ is equivalent to $\boldsymbol{d} \notin \mathcal{Q}$.

Note that, under these assumptions, we can prove that the feasible set defined by (2) is a manifold, see [29, §3.2.1] for more details. This centre, and the characterization of the surface as a manifold, will be used in Section 2.7 to build a fast but inexact projection mapping, referred to as a *quasi-projection*.

The problem defined by (1) is invariant to translations and rotations. Hence, without loss of generality we can consider the following problem in *normal form* [25, Theorem 3.1.1]:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \; & \left\| \boldsymbol{x} - \boldsymbol{x}^0 \right\|_2^2 \\ \text{s.t.} \; & \sum_{i=1}^{n} \lambda_i x_i^2 - 1 = 0, \end{aligned} \tag{3}$$

where $\boldsymbol{\lambda} = \operatorname{spec}(\boldsymbol{B})$, contains the eigenvalues of $\boldsymbol{B}$ sorted in descending order, and $\boldsymbol{x}^0$ is the appropriate transformation of $\tilde{\boldsymbol{x}}^0$. Note that, since the feasible set is nonempty, we have $\lambda_1 > 0$. We will refer to the solution(s) of (3) as the *(true) projection(s)* of $\boldsymbol{x}^0$ onto the quadric.

Note that the centre $\boldsymbol{d}$ is now the origin $\mathbf{0}$. Since this problem is symmetric with respect to the axes, we consider that $\boldsymbol{x}^0 \geqslant 0$, *i.e.*, $\boldsymbol{x}^0$ is located inside the first orthant ($\mathbb{R}^n_+ := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geqslant 0\} = \mathbb{R}^{n,*}_+ \bigcup \{\mathbf{0}\}$).

Remark that if $\lambda_n > 0$, *i.e.*, the quadric is an ellipsoid, and if we have $\sum_{i=1}^n \lambda_i (x_i^0)^2 - 1 > 0$, then the solution of (3) is identical to the solution of

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \; & \left\| \boldsymbol{x}^0 - \boldsymbol{x} \right\|_2^2 \\ \text{s.t.} \; & \sum_{i=1}^{n} \lambda_i x_i^2 - 1 \leqslant 0, \end{aligned}$$

which is a *convex* optimization problem that is easy to solve, *e.g.*, using interior points method (IPM), see [8, 24] for more details, or a black-box commercial solver such as Gurobi [17]. On the other hand, if $\boldsymbol{B}$ is indefinite or if $\sum_{i=1}^n \lambda_i (x_i^0)^2 - 1 < 0$, then we are confronted with a nonconvex optimization problem.

First, let us show that the problem is well-posed, *i.e.*, that there exists a global optimum of (3).

Proposition 2.1. *There exists a global optimum $\boldsymbol{x}^*$ of (3).*

Proof. The objective of (3) is a real-valued, continuous and coercive function defined on a nonempty closed set, therefore there exists a global optimum [6, Theorem 2.32]. □

2.2 KKT conditions

Since the feasible set is nonconvex, the projection operator does not always return a singleton, see [15, Theorem 3.8]. The set of solutions may be a singleton (Figure 1), a finite set (Figure 4), or an infinite set (suppose that $\boldsymbol{x}^0$ is the centre of a sphere, *i.e.*, $\boldsymbol{\lambda} = \mathbb{1}$ and $\boldsymbol{x}^0 = \boldsymbol{0}$). Using the KKT conditions, we can characterize the solutions of (3). The Lagrangian of (3), with Lagrange multiplier μ and with $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{\lambda}) \in \mathbb{R}^{n \times n}$, reads

$$\mathcal{L}(\boldsymbol{x}, \mu) = (\boldsymbol{x} - \boldsymbol{x}^0)^\intercal (\boldsymbol{x} - \boldsymbol{x}^0) + \mu(\boldsymbol{x}^\intercal \boldsymbol{D} \boldsymbol{x} - 1), \tag{4}$$

and the gradient,

$$\boldsymbol{\nabla} \mathcal{L}(\boldsymbol{x}, \mu) = \begin{pmatrix} 2(\boldsymbol{x} - \boldsymbol{x}^0) + 2\mu \boldsymbol{D} \boldsymbol{x} \\ \boldsymbol{x}^\intercal \boldsymbol{D} \boldsymbol{x} - 1 \end{pmatrix}. \tag{5}$$

Note that we can write the i-th equation of (5) as

$$x_i(\mu)(1 + \mu \lambda_i) = x_i^0. \tag{6}$$

Any point $(\boldsymbol{x}, \mu)$ that satisfies

$$\boldsymbol{\nabla} \mathcal{L}(\boldsymbol{x}, \mu) = \boldsymbol{0} \tag{7}$$

is referred to as a *KKT point.* An optimal solution, $\boldsymbol{x}^*$, of (3) must either meet the KKT conditions (4), or fail to satisfy the linear independence constraint qualification (LICQ) criterion. In (3), the latter occurs if $\nabla \Psi(\boldsymbol{x}^*) = \boldsymbol{0}$. This corresponds to the case of a quadric where the centre belongs to the quadric, and is ruled out by the condition $c \neq \frac{\boldsymbol{b}^\intercal \boldsymbol{B}^{-1} \boldsymbol{b}}{4}$.

Isolating $\boldsymbol{x}$ in the first n equations of (7) yields, for $\mu \notin \pi(\boldsymbol{B}) := \left\{-\frac{1}{\lambda} \mid \lambda \text{ is an eigenvalue of } \boldsymbol{B}\right\}$,

$$\boldsymbol{x}(\mu) = (\boldsymbol{I} + \mu \boldsymbol{D})^{-1} \boldsymbol{x}^0, \tag{8}$$

and the $i-$th component can be rewritten as

$$x_i(\mu) = \frac{x_i^0}{1 + \mu \lambda_i}. \tag{9}$$

Note that the set $\pi(\boldsymbol{B})$ corresponds to the *poles* of the rational function (10).

We distinguish two cases:

- Case 1: $\mu \notin \pi(\boldsymbol{B})$. The matrix $(\boldsymbol{I} + \lambda \boldsymbol{D})$ is nonsingular and we have (8).
- Case 2: $\mu = -\frac{1}{\lambda_i}$ for some $i = 1, \ldots, n$. The i-th equation of (7) reads $-2x_i^0 = 0$, therefore μ is a solution only if $x_i^0 = 0$. We first treat the case $\boldsymbol{x}^0 > 0$, denoted as nondegenerate case, in Sections 2.3 and 2.4. Then, the degenerate case $\boldsymbol{x}^0 \geqslant 0$ is tackled in Section 2.5.

In the first case, if we insert (8) in the quadric equation, $\Psi(\boldsymbol{x}) = 0$, we obtain a *scalar*, *extended-real-valued* function

$$\begin{aligned} f : \mathbb{R} \to \overline{\mathbb{R}} : \mu \mapsto f(\mu) &= \Psi(\boldsymbol{x}(\mu)) = \boldsymbol{x}(\mu)^\intercal \boldsymbol{D} \boldsymbol{x}(\mu) - 1 \\ &= \sum_{i=1, x_i^0 \neq 0}^{n} \lambda_i \left(\frac{x_i^0}{1 + \mu \lambda_i} \right)^2 - 1 \end{aligned} \tag{10}$$

of which we want to obtain the roots. Notice that, as the roots correspond to the values μ^* for which $\Psi(\boldsymbol{x}(\mu^*)) = 0$, they can be geometrically understood as the intersections between $\{\boldsymbol{x}(\mu) : \mu \in \mathbb{R}\}$ and the quadric $\mathcal{Q} = \Psi^{-1}(0)$. This is illustrated in Sections 2.3.1 and 2.4.1.

In the following, we show how to efficiently solve (3) by computing a specific root of (10). We first consider the case where $\boldsymbol{x}^0 > 0$ for the ellipsoid case in Section 2.3 and the hyperboloid case in Section 2.4. Then we discuss the case $\boldsymbol{x}^0 \geqslant \mathbf{0}$ in Section 2.5. Finally, we bring everything together into a single algorithm, Algorithm 2, in Section 2.6. We also propose in Section 2.7 a simpler procedure that allows us to map a point to the quadric without having to diagonalize the matrix $\boldsymbol{B}$. As this mapping does not return the true projection, we refer to it as *quasi-projection*.

2.3 Ellipsoid case, $\mathbf{x}^0 > \mathbf{0}$

Here, we assume that the quadric is an ellipsoid, *i.e.*, $\boldsymbol{\lambda} > 0$ and that the initial point lies (strictly) in the first orthant, *i.e.*, $\boldsymbol{x}^0 > \mathbf{0}$.

The goal of this section is twofold. First, we derive several successive results (Propositions 2.2 to 2.5) that characterize the roots of f and the solutions of (3). The combination of these results yields Proposition 2.6 which states that *(3) can be solved by finding the root of f on a given interval $\mathcal{I}$.* Second, we provide a starting point for the Newton root-finding algorithm for efficiently computing this root.

Proposition 2.2. *Under the standing assumptions, every solution $\boldsymbol{x}^*$ of (3) satisfies $\boldsymbol{x}^* > 0$.*

Proof. Recall that any solution of (3) is a KKT point. Using (6) we see that if $(\boldsymbol{x}^*, \mu^*)$ is a KKT point then the positivity of x_i^0 for all $i = 1, \ldots, n$ implies that $x_i^* \neq 0$ for all i.

Let us suppose, for the sake of contradiction, that $\boldsymbol{x}^*$ is a minimizer of (3) and that there exists a nonempty set of indices $J \subseteq \{1, 2, \ldots, n\}$ with $x_j^* < 0$ for all $j \in J$. By symmetry, we can construct $\boldsymbol{x}^{**}$ defined as

$$\boldsymbol{x}^{**} := \begin{cases} x_i^* & \text{if } i \notin J, \\ -x_i^* & \text{if } i \in J, \end{cases}$$

and we have $\Psi(\boldsymbol{x}^{**}) = 0$, *i.e.*, the point belongs to the quadric. The (squared) objective can be computed:

$$\begin{aligned} \left\|\boldsymbol{x}^{**} - \boldsymbol{x}^0\right\|_2^2 &= \sum_{i=1}^{n} (x_i^0 - x_i^{**})^2 \\ &= \sum_{i=1, i \notin J}^{n} (x_i^0 - x_i^*)^2 + \sum_{j \in J} \underbrace{(x_j^0 + x_j^*)^2}_{<(x_j^0 - x_j^*)^2} \\ &< \left\|\boldsymbol{x}^* - \boldsymbol{x}^0\right\|_2^2 \end{aligned}$$

This contradicts the optimality of $\boldsymbol{x}^*$. □

Proposition 2.3. *f, defined as in (10), is strictly decreasing on $\mathcal{I} := \left]-\frac{1}{\lambda_1}, +\infty\right[$.*

Proof. Since $f \in \mathcal{C}^1$ on $\mathcal{I}$, we compute

$$f'(\mu) = -2 \sum_{i=1}^{n} \underbrace{\frac{(\lambda_i x_i^0)^2}{(1+\mu\lambda_i)^3}}_{>0 \text{ for } \mu\in\mathcal{I}}, \tag{11}$$

and this function is negative on $\mathcal{I}$.

□

Proposition 2.4. *Function f restricted to $\mathcal{I}$ has one and only one zero.*

Proof. By Proposition 2.3, f is strictly decreasing, and hence has *at most* one zero. Moreover, $\lim\limits_{\mu\to+\infty} f(\mu) = \Psi(\boldsymbol{d}) = -1 < 0$ and $\lim\limits_{\mu\to-\frac{1}{\lambda_1}} f(\mu) = +\infty$; the continuity of f on $\mathcal{I}$ implies the existence of the zero on $]-\frac{1}{\lambda_1}, +\infty[$. □

Proposition 2.5. *If μ^* is a root of f, and $\mu^* \notin \mathcal{I}$, then $\boldsymbol{x}(\mu^*) \notin \mathbb{R}^n_+$.*

Proof. If $\mu^* \notin \mathcal{I}$, then $\mu^* < -\frac{1}{\lambda_1}$ and therefore the first component of $\boldsymbol{x}(\mu^*)$ reads

$$x_1(\mu^*) = \frac{x_1^0}{1+\mu^*\lambda_1} \tag{12}$$

As the denominator is negative, $\boldsymbol{x}(\mu^*)$ belongs to a different orthant than $\boldsymbol{x}^0$. □

Proposition 2.6. *If $x_i^0 \neq 0$ for all $i = 1, \ldots, n$ and $\boldsymbol{\lambda} > 0$, then the optimal solution of (3) is given by the unique root μ^* of f restricted to $\mathcal{I}$.*

Proof. As shown in Section 2.2, the optimal solution $\boldsymbol{x}^*$ is a KKT point, meaning that it satisfies (7). Using Proposition 2.2, $\boldsymbol{x}^*$ belongs to the same orthant as $\boldsymbol{x}^0$, we are therefore interested in the best KKT solution in the first orthant. However, Proposition 2.5 shows that the corresponding μ^* of the KKT solutions belonging to the same orthant of $\boldsymbol{x}^0$ are located in $\mathcal{I}$ and Proposition 2.4 proves the existence and uniqueness of a root on $\mathcal{I}$, which corresponds therefore to the optimal solution of (3). □

Proposition 2.7. *f is strictly convex on $\mathcal{I}$.*

Proof. $f \in \mathcal{C}^2(\mathcal{I})$ and we compute

$$f''(\mu) = 6 \sum_{i=1}^{n} \frac{(\lambda_i x_i^0)^2 \lambda_i}{(1+\mu\lambda_i)^4},$$

which is positive on $\mathcal{I}$.

□

Proposition 2.8. *Let $\mu^0 \in \mathcal{I} =]-\frac{1}{\lambda_1}, +\infty[$ with $f(\mu^0) > 0$. The Newton-Raphson algorithm with starting point μ^0 converges to μ^*, the unique root of f on $\mathcal{I}$ (as in Proposition 2.6).*

Proof. Let us now prove by induction on k that the sequence $(\mu^k)_{k\in\mathbb{N}}$ provided by Newton's method is an increasing sequence upper bounded by μ^*. The Newton-Raphson iterate for $k = 0, 1, \ldots$ is given by

$$\mu^{k+1} = \mu^k - \frac{f(\mu^k)}{f'(\mu^k)} \,. \tag{13}$$

Using the induction hypothesis, which implies that $f(\mu^k) > 0$ for $\mu^k \neq \mu^*$, and Proposition 2.3, we have

$$\mu^k < \mu^{k+1}.$$

Since f is strictly convex on $\mathcal{I}$ (Proposition 2.7), the tangent of f at a given point is below any chord starting from this point. In particular we have

$$f'(\mu^k) < \frac{f(\mu^k) - f(\mu^*)}{\mu^k - \mu^*},$$

Using the definition of μ^* and rearranging, we obtain

$$\begin{aligned} \mu^k - \frac{f(\mu^k)}{f'(\mu^k)} &< \mu^*, \\ \mu^{k+1} &< \mu^*. \end{aligned}$$

Since the sequence $(\mu^k)_{k\in\mathbb{N}}$ is strictly increasing (for $\mu^k \neq \mu^*$) and bounded, it must converge to a fixed point of (13) which corresponds to a root of f. This concludes the proof as, by Proposition 2.6, there is a unique root of f on $\mathcal{I}$ corresponding to the optimal solution of (3). □

2.3.1 2D example of a nondegenerate projection onto an ellipse

Figure 1 presents an example of a nondegenerate projection, that is with $\boldsymbol{x}^0 > 0$, onto an ellipse. We plot $\boldsymbol{x}(\mu)$, $f(\mu)$, $f'(\mu)$ and $\left\|\boldsymbol{x}(\mu) - \boldsymbol{x}^0\right\|_2$ for μ ranging on $]-\infty, \infty[$. Let us describe how $\boldsymbol{x}(\mu)$, in the top left subfigure, varies when μ decreases from $+\infty$ to $-\infty$. For $\mu \to +\infty$, we have $\boldsymbol{x}(\mu) = \boldsymbol{d}$, where $\boldsymbol{d} = \boldsymbol{0}$ is the quadric centre depicted as a blue dot. Then, while decreasing μ to 0, we reach $\boldsymbol{x}(0) = \boldsymbol{x}^0$. For $\mu \to -\frac{1}{\lambda_1}$, $\boldsymbol{x}(\mu)$ follows an asymptote *and crosses the quadric on* $\boldsymbol{x}(\mu^*)$, the optimal solution of (3), depicted as a purple triangle. Further decreasing μ, $\boldsymbol{x}(\mu)$ reappears on the left part of the asymptote ($x_1 \to -\infty$) and tends to the asymptote ($x_2 \to +\infty$) defined by the other eigenvalue. Finally, $\boldsymbol{x}(\mu)$ converges to the quadric centre when $\mu \to -\infty$, passing again through the quadric in $\boldsymbol{x}(\mu^{**})$, the max point of (3), depicted as a purple square.

The function f is also depicted with its two roots. Note that, depending on the parameters of the problem, it may have one or two additional roots corresponding to local minima or maxima. We also observe in the bottom right figure that $f'(\mu)$ is negative on $\mathcal{I}$, and show the distance to $\boldsymbol{x}^0$ for the different values of $\boldsymbol{x}(\mu)$ in the bottom left figure.

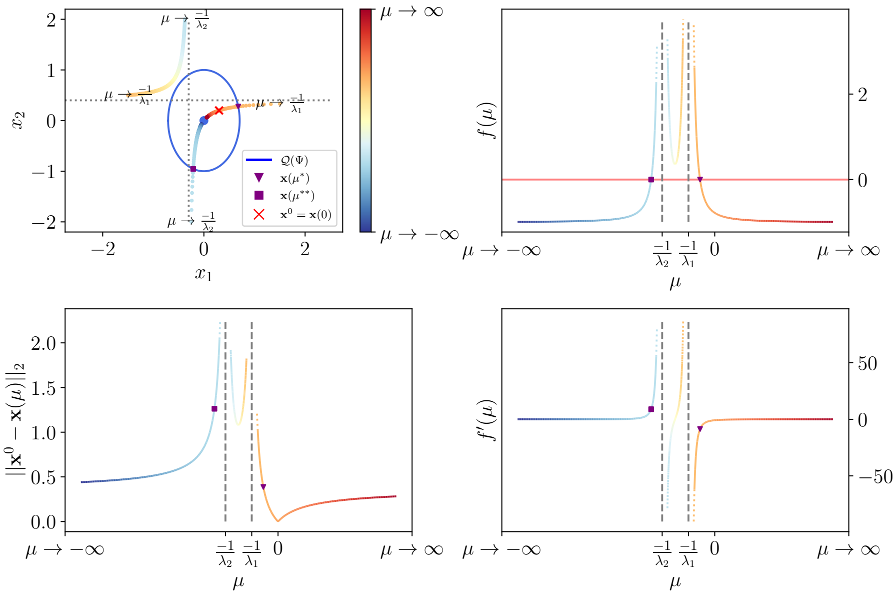


Figure 1: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left|\left|\boldsymbol{x}(\mu) - \boldsymbol{x}^0\right|\right|_2$ for μ ranging on $]-\infty, +\infty[$ in the nondegenerate elliptic case, here $\mathcal{Q}(\Psi) := \mathcal{Q}$.

2.4 Hyperboloid case, $\mathbf{x}^0 > \mathbf{0}$

In the hyperboloid case, there is at least one positive and one negative eigenvalue of $\boldsymbol{B}$. Let $1 \leqslant p \leqslant n-1$ be the number of positive eigenvalues. Let us consider $e_1 := -\frac{1}{\lambda_1}$ and $e_2 := -\frac{1}{\lambda_n}$. We have $0 \in]e_1, e_2[$. We will work analogously as in the ellipsoidal case, with $\mathcal{I} :=]e_1, e_2[$. Proposition 2.2 has no assumption with respect to the positivity of λ, and therefore remains valid. The other propositions can be successively adapted: Proposition 2.9 adapts Proposition 2.3, Proposition 2.10 adapts Proposition 2.4, Proposition 2.11 adapts Proposition 2.5, and finally the main result remains valid, *i.e.*, Proposition 2.12 adapts Proposition 2.6.

We then propose Algorithm 1, also based on Newton-Raphson, to efficiently compute the root of f in $\mathcal{I}$, and hence one of the optimal solutions of (3). An example of the hyperbolic case is provided in Figure 2.

Proposition 2.9. *f, defined as in (10), is strictly decreasing on $\mathcal{I} :=]e_1, e_2[$.*

Proof. Since $f \in \mathcal{C}^1(\mathcal{I})$, we compute

$$f'(\mu) = -2\sum_{i=1}^{p} \frac{(\lambda_i x_i^0)^2}{\underbrace{(1+\mu\lambda_i)^3}_{>0 \text{ if } \mu > e_1}} - 2\sum_{i=p+1}^{n} \frac{(\lambda_i x_i^0)^2}{\underbrace{(1+\mu\lambda_i)^3}_{>0 \text{ if } \mu < e_2}}, \tag{14}$$

and this function is negative on $\mathcal{I}$. □

Proposition 2.10. *Function f restricted to $\mathcal{I}$ with $\boldsymbol{x}^0 > 0$ has one and only one zero.*

Proof. By Proposition 2.9, f is strictly decreasing, and hence has *at most* one zero. Moreover, $\lim\limits_{\mu \to e_1} f(\mu) = +\infty$ and $\lim\limits_{\mu \to e_2} f(\mu) = -\infty$; the continuity of f on $\mathcal{I}$ implies the existence of the zero on $\mathcal{I}$. □

Proposition 2.11. *If μ^* is a root of f, and $\mu^* \notin \mathcal{I}$, then $\boldsymbol{x}(\mu^*) \notin \mathbb{R}_+^n$.*

Proof. If $\mu^* \notin \mathcal{I}$, then either $\mu^* < -\frac{1}{\lambda_1}$ or $\mu^* > -\frac{1}{\lambda_n}$. The first case is already treated in the proof of Proposition 2.5. For the second case, we note that

$$x_n(\mu^*) = \frac{x_n^0}{1+\mu^*\lambda_n}. \tag{15}$$

As $\lambda_n < 0$, the denominator is negative, and thus $\boldsymbol{x}(\mu^*)$ belongs to a different orthant than $\boldsymbol{x}^0$. □

Proposition 2.12. *If $x_i^0 \neq 0$ for all $i = 1, \ldots, n$, then the optimal solution of (3) is given by the unique root μ^* of f restricted to $\mathcal{I} :=]e_1, e_2[$.*

Proof. Since Propositions 2.2, 2.4 and 2.5 are also valid in the hyperboloid case with $\mathcal{I} :=]e_1, e_2[$, the proof is identical to Proposition 2.6. □

Proposition 2.13. *There exists a unique inflexion point, μ^{I}, of f on $\mathcal{I}$.*

Proof. This follows from the monotonicity of $f'' \in \mathcal{C}^\infty(\mathcal{I})$, *i.e.*, $f'''(\mu) < 0$ for all $\mu \in \mathcal{I}$, and because $\lim\limits_{\mu \to e_1^+} f''(\mu) = -\lim\limits_{\mu \to e_2^-} f''(\mu) = +\infty$. □

Since there is a single inflexion point μ^{I}, we can launch in parallel two Newton's algorithms and guarantee that at least one will converge.

Algorithm 1 Double Newton

```
if $f(0) < 0$ then
    Use bisection method (see [10, Chapter 2.1]) to find $\mu_{\mathrm{s}} \in ]e_1, 0[$ s.t. $f(\mu_{\mathrm{s}}) > 0$
else if $f(0) > 0$ then
    Use bisection method to find $\mu_{\mathrm{s}} \in ]0, e_2[$ s.t. $f(\mu_{\mathrm{s}}) < 0$
else
    return 0
end if
$\mu_0 \leftarrow$ Newton(0) {This and the next line are run in parallel}
$\mu_1 \leftarrow$ Newton($\mu_{\mathrm{s}}$)
return $\mu_0, \mu_1$ {Returns the output solution of the first of the two parallel Newtons that is finished}
```

Proposition 2.14. *One of the two Newton's methods of Algorithm 1 converges to μ^*, with $\boldsymbol{x}(\mu^*)$ (defined in (8)) the optimal solution of (3).*

Proof. This proof relies on the double initiation of Newton's method in Algorithm 1: one starting from a positive value, and the other from a negative value. We comment on the sign of $f(\mu^{\mathrm{I}})$:

- If $f(\mu^{\mathrm{I}}) < 0$, then $\mu^* \in]e_1, \mu^{\mathrm{I}}[$ and the function is convex on this interval. The situation is similar to Proposition 2.8, and any starting point μ_{s} with $f(\mu_{\mathrm{s}}) > 0$ is a valid starting point, in the sense that the sequence of iterates converges to μ^*.
- If $f(\mu^{\mathrm{I}}) > 0$, then $\mu^* \in]\mu^{\mathrm{I}}, e_2[$ and the function is concave on this interval. Using a similar argument as Proposition 2.8, any starting point μ_{s} with $f(\mu_{\mathrm{s}}) < 0$ is a valid starting point.
- If $f(\mu^{\mathrm{I}}) = 0$, any starting point in $\mathcal{I}$ is a valid starting point.

□

Remark that with the knowledge of the value of μ^{I}, we could launch a single Newton scheme with the appropriate starting point. Unfortunately, computing μ^{I} amounts to computing the root of f'' which is at least as costly as finding the root of f.

2.4.1 2D example of a nondegenerate projection onto a hyperbola

Figure 2 shows an example of a nondegenerate projection onto a hyperbola. We observe a similar image of $\boldsymbol{x}(\mu)$, with two asymptotes. We see

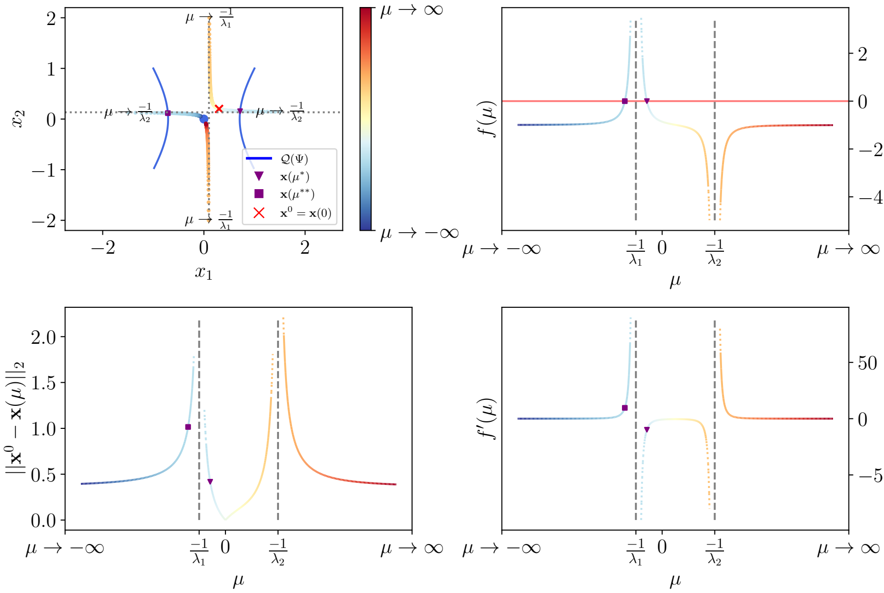


Figure 2: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left\|\boldsymbol{x}(\mu) - \boldsymbol{x}^0\right\|_2$ for μ ranging on $]-\infty, +\infty[$ in the nondegenerate hyperbolic case, here $\mathcal{Q}(\Psi) := \mathcal{Q}$.

that $f(\mu)$ has a unique inflexion point on $\mathcal{I}$. In this example, the inflexion point is on the right of the root, and thus we know that starting a Newton-Raphson scheme in some μ_s with $f(\mu_\mathrm{s}) > 0$ yields a sequence that converges to μ^*.

2.5 Degenerate case, $\mathrm{x}^0 \geqslant 0$

Let us first assume that all eigenvalues of $\boldsymbol{B}$ are distinct, the case with repeated eigenvalues is treated at the end of the current section.

2.5.1 All eigenvalues are distinct

The discussion is largely similar to Section 2.4, with the following differences: i) f is continuous at $-\frac{1}{\lambda_i}$ if the associated component of $\boldsymbol{x}^0$ is equal to zero and ii) at most two additional KKT points can be obtained for each component of $\boldsymbol{x}^0$ that is equal to zero.

We tackle these issues as follows. First, we arbitrarily decide to single out a solution in the first orthant. Second, we change the definition of e_1, e_2, to account for the continuity of f in $-\frac{1}{\lambda_i}$: $\lim_{\mu \to -\frac{1}{\lambda_i}} f(\mu) \neq \infty$ if $x_i^0 = 0$. Finally, we show how to analytically compute these additional solutions.

Let $I := \{1, 2, \ldots, n\}$ and $K \subseteq I$, it is clear that Proposition 2.2 is not valid any more if $|K|$ entries of $\boldsymbol{x}^0$ are equal to zero. Indeed, if $\boldsymbol{x}^*$

is an optimal solution that belongs to the same orthant as $\boldsymbol{x}^0$, then $\boldsymbol{x}^{**}$ defined as follows

$$x_i^{**} = \begin{cases} x_i^* & \forall i \notin K, \\ -x_i^* & \forall i \in K, \end{cases} \tag{16}$$

is also an optimal solution. In fact, up to $2^{|K|} - 1$ solutions outside the first orthant can be obtained by mirroring $\boldsymbol{x}^*$ along a selected set of components in K.

As we are interested in finding *one* of the optimal solutions, we note that we can also restrict our search to the first orthant.

Proposition 2.15. *Given $\boldsymbol{x}^0 \geqslant \mathbf{0}$, there exists an optimal solution $\boldsymbol{x}^*$ of (3) such that $\boldsymbol{x}^* \geqslant \mathbf{0}$.*

Proof. Let $\boldsymbol{x}^*$ be an optimal solution. The existence of $\boldsymbol{x}^*$ follows from Proposition 2.1. Using a similar argument as in the proof of Proposition 2.2, we have $\operatorname{sign}(x_j^0) = \operatorname{sign}(x_j^*) \quad \forall j \in I \backslash K$. Let

$$x_i^{**} = \begin{cases} -x_i^* & \forall i \in K \text{ with } \operatorname{sign}(x_i^*) \neq \operatorname{sign}(x_i^0), \\ x_i^* & \text{elsewhere.} \end{cases}$$

This feasible point has the same objective as $\boldsymbol{x}^*$ and is located in the same orthant as $\boldsymbol{x}^0$. □

For $\mu \notin -\frac{1}{\operatorname{spec}(\boldsymbol{B})}$, we change the definition of e_1, e_2 as

$$\begin{aligned} e_1 &= \max_{\{i \in I \mid \lambda_i > 0, x_i^0 \neq 0\}} -\frac{1}{\lambda_i} \\ e_2 &= \min_{\{i \in I \mid \lambda_i < 0, x_i^0 \neq 0\}} -\frac{1}{\lambda_i} \end{aligned} \tag{17}$$

and $e^1 = -e^2 := -\infty$ if the max or min is empty. This takes into account the continuity of f at $\mu = -\frac{1}{\lambda_i}$ if $x_i^0 = 0$ for some index i.

Let us adapt Propositions 2.9 to 2.11 to the degenerate case.

Proposition 2.16. *f, defined as in (10), is strictly decreasing on $\mathcal{I} :=]e_1, e_2[$.*

Proof. Since $f \in \mathcal{C}^1(\mathcal{I})$, we compute

$$f'(\mu) = -2 \sum_{i=1, i \notin K}^{p} \frac{(\lambda_i x_i^0)^2}{\underbrace{(1 + \mu\lambda_i)^3}_{>0 \text{ if } \mu > e_1}} - 2 \sum_{i=p+1, i \notin K}^{n} \frac{(\lambda_i x_i^0)^2}{\underbrace{(1 + \mu\lambda_i)^3}_{>0 \text{ if } \mu < e_2}}, \tag{18}$$

and this function is negative on $\mathcal{I}$. □

Proposition 2.17. *Function f restricted to $\mathcal{I}$ has one and only one zero if $\exists i \in I^+ := \{i \in I \mid \lambda_i > 0\}$ with $x_i^0 \neq 0$.*

Proof. We note that the technical assumption on x_i^0 ensures that $\lim\limits_{\mu \to e_1} f(\mu) > 0$. For $\lim\limits_{\mu \to e_2} f(\mu)$ we distinguish two cases:

- either $e_2 = +\infty$ and $\lim\limits_{\mu \to e_2} = -1$;
- or $e_2 = \min\limits_{i \in I \,|\, \lambda_i < 0, x_i^0 \neq 0} -\frac{1}{\lambda_i}$ and $\lim\limits_{\mu \to e_2} f(\mu) = -\infty$.

Since in both cases the limit is negative, and f is continuous and strictly decreasing on $\mathcal{I}$, there exists a unique zero on this interval. □

Proposition 2.18. *If μ^* is a root of f, and $\mu^* \notin \mathcal{I}$, then $\boldsymbol{x}(\mu^*) \notin \mathbb{R}^n_+$.*

Proof. If $\mu^* \notin \mathcal{I}$, then either $e_1 \neq -\infty$ and $\mu < e_1$ or $e_2 \neq +\infty$ and $\mu > e_2$.

The proof follows from the definition of e_i, *e.g.*, in the first case we note that

$$x_{i_1}(\mu^*) = \frac{x_{i_1}^0}{1 + \mu\lambda_{i_1}},$$

where $i_1 = \operatorname{argmax}_{\{i \in I \,|\, \lambda_i > 0, x_i^0 \neq 0\}} -\frac{1}{\lambda_i}$. This implies that $\boldsymbol{x}(\mu^*)$ belongs to a different orthant that $\boldsymbol{x}^0$ since the numerator is nonzero and the denominator is negative. □

Remark that if $x_i^0 = 0 \quad \forall i \in I^+$, meaning that the assumption on $\boldsymbol{x}^0$ of Proposition 2.17 does not hold, then $f(\mu)$ reads

$$f(\mu) = \sum_{i \in I^-} \lambda_i \left(\frac{x_i^0}{1 + \mu\lambda_i} \right)^2 - 1,$$

where $I^- := I \backslash I^+ = \{p+1, p+2, \ldots, n\}$. This function is negative on $\mathbb{R}$. In this specific case, f *does not provide any KKT point*, such a situation is depicted in Figure 5. However, the problem is solvable, due to additional KKT points that appear when $\boldsymbol{x}^0$ is located on the axes.

Indeed, if $\mu = -\frac{1}{\lambda_k}$ for $k \in K$ then the $k-$th entry of (7) reads

$$2(x_k - x_k^0) + 2\mu\lambda_k x_k = 0 \tag{19}$$

which is true no matter x_k. Therefore, we obtain at most *two additional* solutions of the Lagrangian system (7). Geometrically, this corresponds to looking at the intersection between i) a line perpendicular to the axis corresponding to the component k where $x_k^0 = 0$ and ii) the quadric. These solutions, if they exist, can be computed as

$$(\boldsymbol{x}_k^{\mathrm{d}})_i = \begin{cases} \frac{x_i^0}{1 - \frac{\lambda_i}{\lambda_k}} & \text{if } i \neq k, \\ \pm\sqrt{\frac{1}{\lambda_k}\left(1 - \sum\limits_{j \in I, j \neq k} \lambda_j \left(\frac{x_j^0}{1 - \frac{\lambda_j}{\lambda_k}} \right)^2 \right)} & \text{if } i = k, \end{cases} \tag{20}$$

where $(\cdot)_i$ selects the $i-$th component, and we choose the "+" solution that lies in the first orthant.

Such a situation is depicted in Figure 3. We observe that $\boldsymbol{x}(\mu)$ moves around the axis corresponding to the component of $\boldsymbol{x}^0$ which is equal to zero. Moreover, the additional solution is depicted in green in Figs. 3

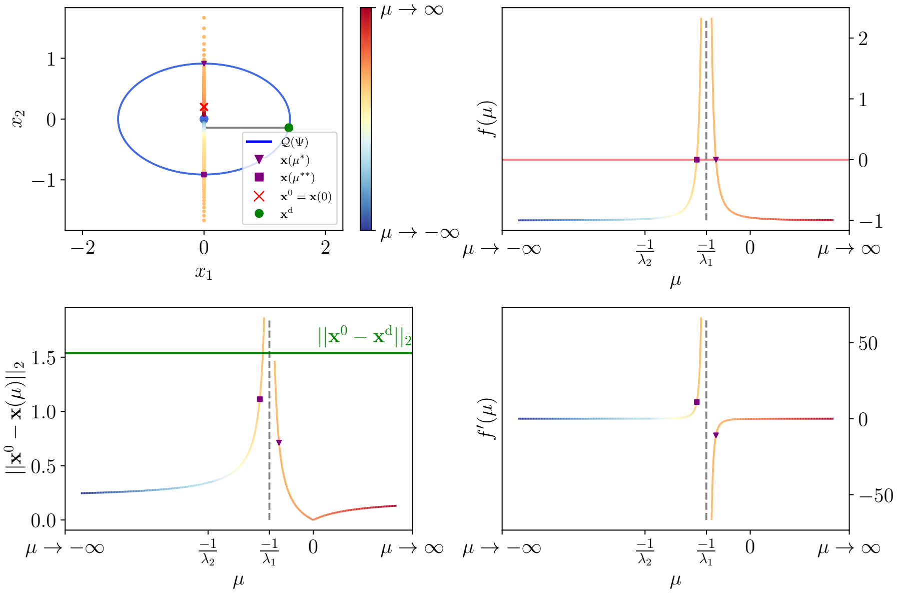


Figure 3: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left|\left|\boldsymbol{x}(\mu)-\boldsymbol{x}^0\right|\right|_2$ for μ ranging on $]-\infty,+\infty[$ in the degenerate elliptic case. The optimal solution is the root of f and not $\boldsymbol{x}_2^{\mathrm{d}}$: the green line showing $\left|\left|\boldsymbol{x}^0-\boldsymbol{x}_2^{\mathrm{d}}\right|\right|$ is above the purple triangle in the lower left figure.

and 4. Note that in Figure 3, the optimal solution is a root of f and in Figure 4, it is the additional solution.

Remark that there is no intersection, and therefore no additional solution to (7), if $\frac{1}{\lambda_k}\left(1-\sum_{j\in I, j\neq k}\lambda_j\left(\frac{x_j^0}{1-\frac{\lambda_j}{\lambda_k}}\right)^2\right)<0$, see, *e.g.*, Figure 6.

2.5.2 2D examples of degenerate projections

Figs. 3 and 4 show two examples of *degenerate* projections onto an ellipse. Figure 3 depicts an example where the optimal solution is given by the KKT point corresponding to the root of f. Figure 4 depicts an example where the optimal solution is given by the KKT point corresponding to $\mu=-\frac{1}{\lambda_2}$. Notice that in these (degenerate) cases, one of the asymptotes of the $\boldsymbol{x}(\mu)$ image disappears, and the image is hence along one of the axes. Moreover, one of the discontinuities of f, f' and $\left|\left|\boldsymbol{x}(\mu)-\boldsymbol{x}^0\right|\right|_2$ disappears as, *e.g.*, $\lim_{\mu\to-\frac{1}{\lambda_k}} f(\mu)=\sum_{i=1,i\neq k}^{n}\lambda_i\left(\frac{x_i^0}{1-\frac{\lambda_i}{\lambda_k}}\right)^2-1\neq\infty$.

Figs. 5 and 6 show two examples of *degenerate* projections onto a hyperbola. Figure 5 depicts an example where f has no root. This is not

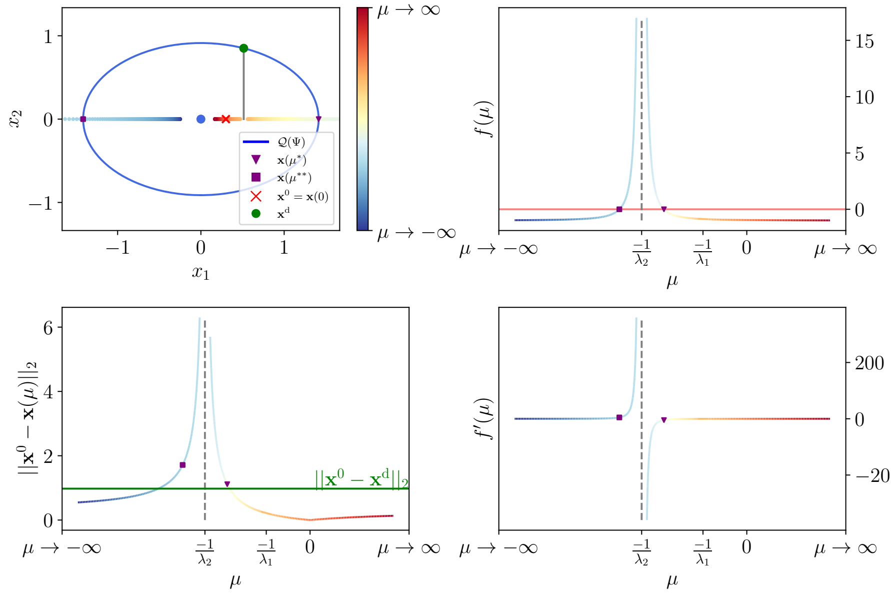


Figure 4: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left\|\boldsymbol{x}(\mu)-\boldsymbol{x}^0\right\|_2$ for μ ranging on $]-\infty,+\infty[$ in the degenerate elliptic case. The optimal solution is not the root of f but $\boldsymbol{x}_1^{\mathrm{d}}$: the green line showing $\left\|\boldsymbol{x}^0-\boldsymbol{x}_1^{\mathrm{d}}\right\|_2$ is below the purple triangle.

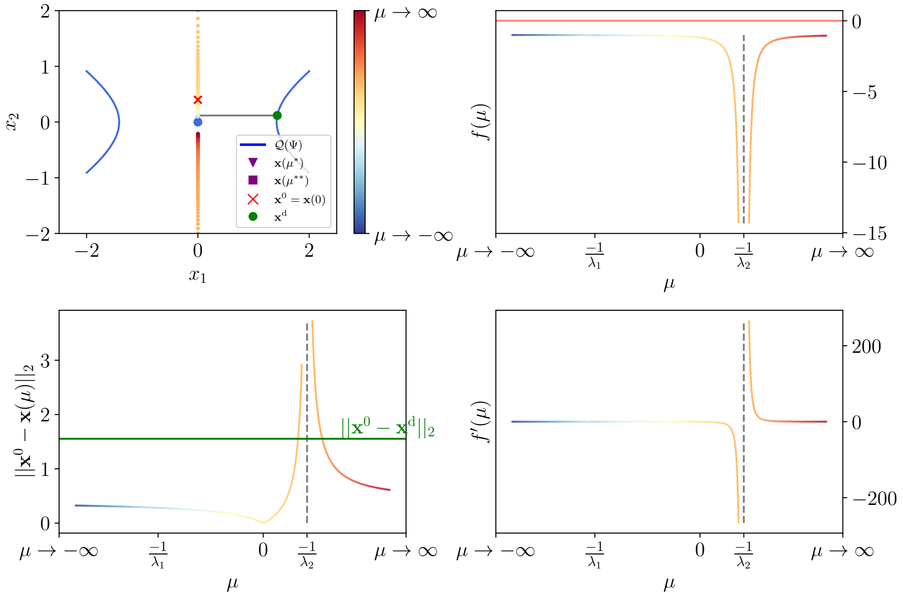


Figure 5: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left\|\boldsymbol{x}(\mu) - \boldsymbol{x}^0\right\|_2$ for μ ranging on $]-\infty, +\infty[$ in the degenerate hyperbolic case. The optimal solution is $\boldsymbol{x}_1^{\mathrm{d}}$, as f has no root.

an issue because the optimal solution is given, in this case, as one of the $\boldsymbol{x}_k^{\mathrm{d}}$ depicted in green which are derived in (20). Figure 6 shows an example where there is no intersection between the grey line and the quadric, and therefore no $\boldsymbol{x}_k^{\mathrm{d}}$. This is not an issue because then there must exist a root μ^* of f in the first orthant, which is the optimal solution (purple triangle). Remark that, if $\sum_{i=1}^{n} \lambda_i (x_i^0)^2 - 1 > 0$, then both $\boldsymbol{x}(\mu^*)$ and $\boldsymbol{x}_k^{\mathrm{d}}$ are KKT points, and one of them is the optimal solution.

2.5.3 Some eigenvalues are repeated

Let $\overline{\boldsymbol{\lambda}}$ be the vector of the unique eigenvalues of $\boldsymbol{B}$, sorted in descending order, let $k = 1, \ldots, \left|\overline{\boldsymbol{\lambda}}\right|$ be a given component of $\overline{\boldsymbol{\lambda}}$, let L_k be a subset of I corresponding to the same eigenvalue, *i.e.*, $L_k := \left\{ l \in I \mid \lambda_l = \overline{\lambda}_k \right\}$, and let K_k be a subset of L_k where the associated component of $\boldsymbol{x}^0$ is equal to zero, *i.e.*,

$$K_k := \left\{ i \in I \mid \lambda_i = \overline{\lambda}_k, x_i^0 = 0 \right\}.$$

Proposition 2.19. *Let L_k and K_k be defined as above. There exists a solution of (7) with $\mu^* = -\frac{1}{\overline{\lambda}_k}$ only if $L_k = K_k$.*

Proof. Let $(\boldsymbol{x}^*, \mu^*)$ be a solution of (7) with $\mu^* = -\frac{1}{\overline{\lambda}_k}$. Let us assume,

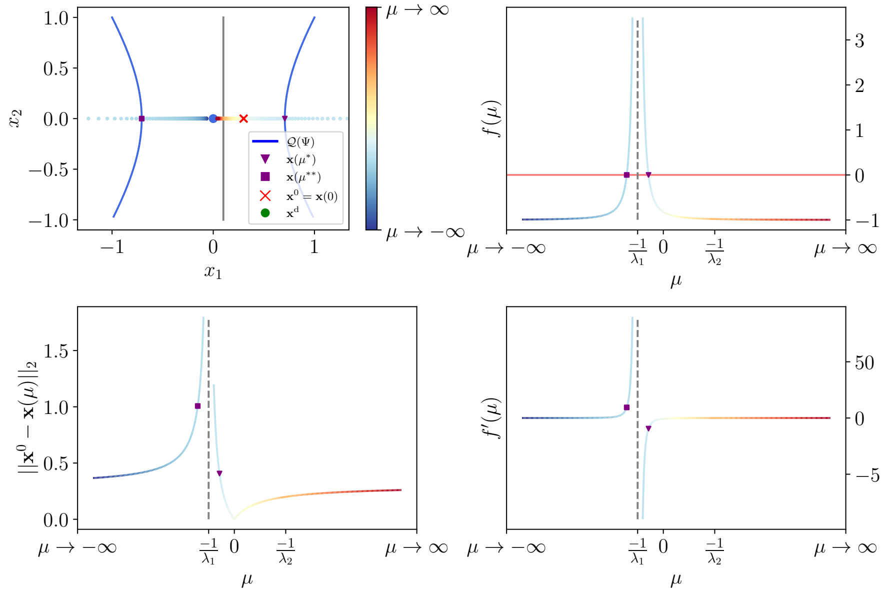


Figure 6: Image of $\boldsymbol{x}(\mu)$ and graphs of $f(\mu)$, $f'(\mu)$, $\left\|\boldsymbol{x}(\mu) - \boldsymbol{x}^0\right\|_2$ for μ ranging on $]-\infty, +\infty[$ in the degenerate hyperbolic case. The optimal solution is $\boldsymbol{x}(\mu^*)$, the root of f. There is no additional KKT point $\boldsymbol{x}_k^{\mathrm{d}}$; the grey line in the upper left panel has no intersection with the quadric.

for the sake of contradiction, that $L_k \neq K_k$, or equivalently, that $\exists i \in I$ with $\lambda_i = \overline{\lambda}_k$ but $x_i^0 \neq 0$. The $i-$th component of (7) reads

$$2(x_i - x_i^0) - 2x_i = 0,$$

which does not hold. □

Remark that Proposition 2.19 is a left implication and it is possible that no solution of (7) exists with $\mu^* = -\frac{1}{\overline{\lambda}_k}$ and $L_k = K_k$, see, *e.g.*, Figure 5.

If $|K_k| = 1$, the discussion is analogous to the previous paragraph: at most two KKT solutions are obtained as the intersection between a line and the quadric, but for $|K_k| > 1$, we have to take the intersection between a plane $\pi := \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid x_{k'} = 0 \, \forall k' \in K_k \right\}$, and the quadric. Geometrically, the intersection—if there is one, *i.e.*, if the argument of the square root below is positive—will be a $|K_k| - 1$ hypersphere, in the corresponding subspace of $\mathbb{R}^n$:

$$\pi \bigcap \mathcal{Q} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \text{ s.t } x_i = \frac{x_i^0}{1 - \frac{\lambda_i}{\overline{\lambda}_k}} \text{if } i \notin K_k, \right.$$
$$\left. \sum_{i \in K_k} x_i^2 = \frac{1}{\overline{\lambda}_k} \left(1 - \sum_{j \in I \setminus K_k} \lambda_j \left(\frac{x_j^0}{1 - \frac{\lambda_j}{\overline{\lambda}_k}} \right)^2 \right) \right\} \tag{21}$$

and every point belonging to this hypersphere is a KKT point. Moreover, all the points in this hypersphere achieve the same value for the objective function of (3). Hence, for the purpose of finding one of the optimal solutions of (3), we can keep in our list of candidates just one element of (21). In particular, we can arbitrarily select *one* solution that lies in the first orthant by setting to zero all components of K_k except one (k'):

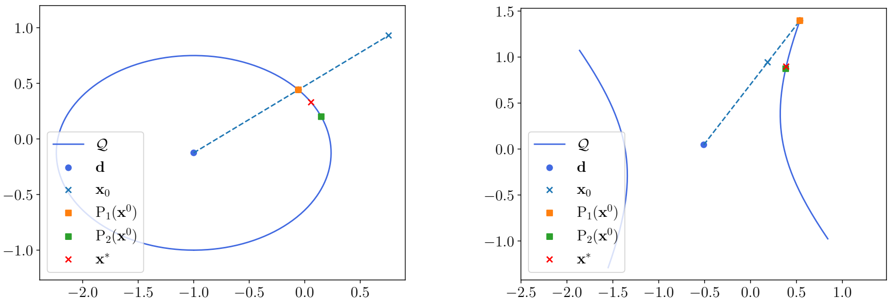
$$(\boldsymbol{x}_k^{\mathrm{d}})_i = \begin{cases} \frac{x_i^0}{1 - \frac{\lambda_i}{\overline{\lambda}_k}} & \text{if } i \notin K_k, \\ \sqrt{\frac{1}{\overline{\lambda}_k} \left(1 - \sum_{j \in I \setminus K_k} \lambda_j \left(\frac{x_j^0}{1 - \frac{\lambda_j}{\overline{\lambda}_k}} \right)^2 \right)} & \text{if } i = k', \\ 0 & \text{if } i \in K_k, i \neq k' \end{cases} \tag{22}$$

Any $k' \in K_k$ works, let us choose without loss of generality $k' := \min\limits_{i \in K_k} i$. As a matter of fact, this is equivalent to restricting the search to the subspace $\{\boldsymbol{x} \in \mathbb{R}^n \mid x_i = 0 \quad \forall i \in K_k \setminus \{k'\}\}$, because all solutions of the hypersphere have the same objective. In this subspace, the problem is analogous to the case $|K_k| = 1$, *i.e.*, the intersection between a line and a quadric.

2.6 Bringing everything together

Let us give a full characterization of an optimal solution to (3).

Proposition 2.20. *There is an optimal solution of (3) in the set* $\{\boldsymbol{x}(\mu^*)\} \bigcup X^d$ *where*



(a) Illustration of the exact and quasi-projections for an ellipse.

(b) Illustration of the exact and quasi-projections for an hyperbola.

Figure 7: Comparison between the exact and quasi-projections onto a 2D quadric.

- $\boldsymbol{x}(\mu^*)$ *is defined by (8), where* μ^* *is the unique root of* f *on* $\mathcal{I} =]e_1, e_2[$*, and* e_1*,* e_2 *are given by (17);*
- $\boldsymbol{X}^{\mathrm{d}} := \left\{ \boldsymbol{x}_k^{\mathrm{d}} \mid k = 1, \dots, \left|\overline{\boldsymbol{\lambda}}\right|, \text{and } |K_k| > 0 \right\}$ *as defined in Section 2.5.*

Proof. Since the quadric is central, no point fulfils the LICQ condition. Hence, the solution of (3)— which exists by Proposition 2.1— must be a KKT point. The KKT points are the solutions of (7), *i.e.*,

$$x_i(1 + \mu\lambda_i) = x_i^0 \quad \text{for all } i \in I \text{ and } \sum_{i \in I} \lambda_i x_i^2 = 1$$

Hence $(\boldsymbol{x}^*, \mu^*)$ is a solution of the KKT conditions (7) if and only if one of the following holds:

(i) $\mu^* \neq -\frac{1}{\lambda_i}$ for all i, μ^* is a root of f defined in (10), and $\boldsymbol{x}^*$ satisfies (8).

(ii) $\mu^* = -\frac{1}{\lambda_k}$ for some k, $x_i^0 = 0$ for all i such that $\lambda_i = \lambda_k$, and, letting K_k be the set of those i's, $\boldsymbol{x}^* \in \pi \bigcap \mathcal{Q}$ defined in (21)

In case (i), we have seen that the smallest objective of (3) is given by the— possibly nonexistent—unique root of f. In case (ii), all the points in (21)—which may be empty—achieve the same value for the objective of (3), and (22)—defined if and only if (21) is nonempty—is one of those points. Finally, recall that Proposition 2.1 proves the existence of one optimal solution to (3): either case (i) or case (ii) will provide a solution. □

The full procedure to compute the projection of any point $\boldsymbol{x}^0$ onto a nonempty and non-cylindrical central quadric is given in Algorithm 2.

Algorithm 2 Exact projection onto a non-cylindrical central quadric in normal form

```
Require: $\boldsymbol{\lambda}$, the eigenvalues corresponding to (3), and $\boldsymbol{x}^0 \in \mathbb{R}^n_+$
  $e_1 \leftarrow \max\left\{\max_{\{i\in I \mid \lambda_i>0, x_i^0\neq 0\}} -\frac{1}{\lambda_i}, -\infty\right\}$ {If the inner max is empty, $e_1 = -\infty$}
  $e_2 \leftarrow \min\left\{\min_{\{i\in I \mid \lambda_i<0, x_i^0\neq 0\}} -\frac{1}{\lambda_i}, +\infty\right\}$ {If the inner min is empty, $e_2 = +\infty$}

  $\boldsymbol{D} \leftarrow \mathrm{diag}(\boldsymbol{\lambda})$
  $\boldsymbol{x}(\mu) \leftarrow (\boldsymbol{I} + \mu\boldsymbol{D})^{-1}\boldsymbol{x}^0$
  $f(\mu) \leftarrow \sum_{i=1, x_i^0\neq 0}^{n} \lambda_i \left(\frac{x_i^0}{1+\mu\lambda_i}\right)^2 - 1$
  if $e_1 \neq -\infty$ then
    if $e_2 = +\infty$ then
      $\mu_0 \leftarrow \mathrm{bisection}(f, e_1)$ {Use bisection search (see [10, Chapter 2.1]) to find $\mu^0 > e_1$ with $f(\mu^0) > 0$}
      $\mu^* \leftarrow \mathrm{root}(f, \mu_0)$ {Using Newton with starting point $\mu^0$}
    else
      $\mu^* \leftarrow \mathrm{root}(f, e_1, e_2)$ {Using Algorithm 1}
    end if
  end if{See contrapositive of Proposition 2.17: $e_1 = -\infty \Rightarrow f$ has no root on $\mathcal{I}$}
  $\overline{\boldsymbol{\lambda}} \leftarrow \mathrm{unique}(\boldsymbol{\lambda})$
  $\boldsymbol{X}^{\mathrm{d}} \leftarrow [\,]$
  for $k = 1, \dots, |\overline{\boldsymbol{\lambda}}|$ do
    $K_k \leftarrow \left\{i \in I \mid \lambda_i = \overline{\lambda}_k, x_i^0 = 0\right\}$
    $L_k \leftarrow \left\{i \in I \mid \lambda_i = \overline{\lambda}_k\right\}$
    if $K_k = L_k$ and $\frac{1}{\overline{\lambda}_k}\left(1 - \sum_{j\in I\setminus K_k} \lambda_j \left(\frac{x_j^0}{1-\frac{\lambda_j}{\overline{\lambda}_k}}\right)^2\right) > 0$ then
      $\boldsymbol{x}_k^{\mathrm{d}} \leftarrow (22)$
      $\boldsymbol{X}^{\mathrm{d}}.\mathrm{append}(\boldsymbol{x}_k^{\mathrm{d}})$
    end if
  end for
  return $\underset{\boldsymbol{x}\in\{\boldsymbol{x}(\mu^*)\}\bigcup \boldsymbol{X}^{\mathrm{d}}}{\arg\min} \left\|\boldsymbol{x}^0 - \boldsymbol{x}\right\|_2$ {The min is taken over at most $n+1$ values}
```

2.7 Quasi-projection onto the quadric

The procedure detailed in Algorithm 2 is an exact projection, but it requires computing the full eigenvalue decomposition of $\boldsymbol{B}$, including the eigenvectors, which may be expensive for problems of large dimension. In this subsection, we detail a geometric procedure which allows us, under some conditions (see Section 2.7.1), to map a given point to the feasible set $\mathcal{Q}$ of (1). We refer to such a mapping as a *quasi-projection*.

Definition 2.1. Quasi-projection. Let $\boldsymbol{x} \in \mathbb{R}^n$, a *quasi-projection* on the quadric $\mathcal{Q}$ is a mapping

$$\mathrm{P} : \mathcal{D} \to \mathcal{Q} : \boldsymbol{x} \mapsto \mathrm{P}(\boldsymbol{x}) \,,$$

where $\mathcal{D}$ is a nonempty subset of $\mathbb{R}^n$.

Note that this definition is broad, and includes the projection operator. Ideally, $\mathcal{D}$ should be $\mathbb{R}^n$, but we allow the quasi-projection to fail to map some points.

This quasi-projection is inspired from the retraction in [7, 29], and from the following observation: the projection of a given point $\boldsymbol{x}$ onto a sphere that can be analytically computed by looking at the intersection between the sphere and the half line defined by the sphere centre and $\boldsymbol{x}$. As the quadric $\mathcal{Q}$ is by assumption a *central quadric*, it is tempting to approximate the projection by the same mechanism described above for the sphere. This yields a first variant of the *quasi-projection*. The second variant is obtained by searching for the intersection between the quadric and the line passing through $\boldsymbol{x}^0$ along the direction $\nabla\Psi(\boldsymbol{x}^0)$.

We are looking at the intersection between the quadric and the line starting from $\boldsymbol{x}^0$ along some direction $\boldsymbol{\xi}$. The intersections are parametrized as $\boldsymbol{x}^0 + \beta\boldsymbol{\xi}$, where β satisfies $\Psi(\boldsymbol{x}^0 + \beta\boldsymbol{\xi}) = 0$, or equivalently, $b_1\beta^2 + b_2\beta + b_3 = 0$, for appropriate b_i's. The b_i's are given in Algorithm 3, see [7, §3.2] for more details. In order to select the point that is the closest to $\boldsymbol{x}^0$ among both intersections, the β which is the closest to zero is chosen.

We detail two variants of our *quasi-projection*:

- $\boldsymbol{\xi} = \boldsymbol{d} - \boldsymbol{x}^0$: this is analogous to the retraction used in [7], is referred to as *centre-based* quasi-projection, and is denoted by P_1;
- $\boldsymbol{\xi} = \nabla\Psi(\boldsymbol{x}^0) = 2\boldsymbol{B}\boldsymbol{x}^0 + \boldsymbol{b}$: the direction is given by the gradient of the level curve of Ψ at $\boldsymbol{x}^0$. We refer to it as *gradient-based* quasi-projection and denote it as P_2.

The quasi-projection procedure is given in Algorithm 3 and depicted in Figure 7 for both strategies.

2.7.1 Failure of the quasi-projection

Remark that for $\boldsymbol{x}^0 = \boldsymbol{0}$, P_1 is not defined, and for the hyperboloid case, the set where P_1 is not defined ($\mathbb{R}^n \backslash \mathcal{D}$), is a closed set including $\boldsymbol{0}$. Examples of nontrivial points that cannot be mapped using P_1 are provided in Figure 8. Indeed, in these cases, there is no intersection between the quadric and the line starting from $\boldsymbol{x}^0$. We tackle this issue by resorting to the exact projection from Algorithm 2 whenever this situation occurs.

Algorithm 3 Quasi-projection onto the quadric

```
Require: $\boldsymbol{x}^0 \in \mathbb{R}^n$, a non-cylindrical central quadric $\mathcal{Q}$ with centre $\boldsymbol{d}$, a direction $\boldsymbol{\xi}$
  $b_1 \leftarrow \boldsymbol{\xi}^\intercal \boldsymbol{B} \boldsymbol{\xi}$
  $b_2 \leftarrow 2\boldsymbol{x}^{0\intercal} \boldsymbol{B} \boldsymbol{\xi} + \boldsymbol{b}^\intercal \boldsymbol{\xi}$
  $b_3 \leftarrow \boldsymbol{x}^{0\intercal} \boldsymbol{B} \boldsymbol{x}^0 + \boldsymbol{b}^\intercal \boldsymbol{x}^0 + c$
  if $b_2^2 - 4b_1b_3 < 0$ then
    return None {it may happen that no intersection is available see, e.g., Figure 8}
  else
    $\Delta \leftarrow \sqrt{b_2^2 - 4b_1b_3}$
    $\beta^+ \leftarrow \frac{-b_2+\Delta}{2b_1}$
    $\beta^- \leftarrow \frac{-b_2-\Delta}{2b_1}$
    if $b_2 > 0$ then
      $\beta \leftarrow \beta^+$
    else
      $\beta \leftarrow \beta^-$
    end if{we select the closest to $\boldsymbol{x}^0$ of the two intersections}
    return $\boldsymbol{x}^0 + \beta\boldsymbol{\xi}$
  end if
```

There are points $\boldsymbol{x}^0$ for which P_2 returns None, but it returns a point when $\boldsymbol{x}^0$ is close enough to $\mathcal{Q}$.

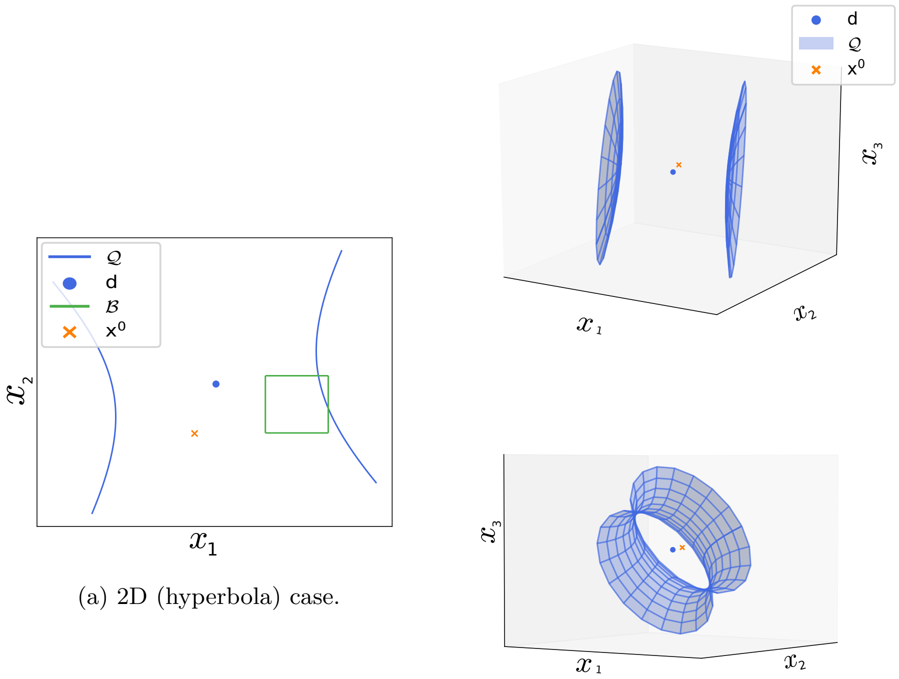
2.7.2 Features of the quasi-projection

In general the quasi-projection is not exact, in the sense that the resulting point is *not* the optimal solution of (1). However, we expect the quasi-projection to be close to optimality when the point is close enough to the quadric. Such a behaviour is observed in our simulations in Section 4.2. Also, in the specific case when the quadric is a sphere, then both P_1 and P_2 solve (1).

3 Splitting methods for the projection onto the intersection of a box and a quadric

This section is devoted to the analysis of the projection onto a feasible set, Ω, which is the intersection between a box and a non-cylindrical central quadric. Let $\mathcal{B}$ be a nonempty n-dimensional hyper-cube or *box*, aligned with the axes:

$$\mathcal{B} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \underline{x}_i \leqslant x_i \leqslant \overline{x}_i,\ \forall i = 1 \dots n\}, \tag{23}$$



(a) 2D (hyperbola) case.

(b) 3D (two and one-sheet hyperboloid) cases.

Figure 8: Illustration of a failure of the quasi-projection (P_1): the point $\boldsymbol{x}^0$ cannot be mapped to the quadric using Algorithm 3. Indeed, the line defined by $\boldsymbol{x}^0$ and $\boldsymbol{d}$ does not intersect with the quadric.

for given lower and upper bounds $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$, and $\mathcal{Q}$ a quadric. The optimization problem at hand reads

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} & \left\|\boldsymbol{x}-\boldsymbol{x}^0\right\|_2^2 \\ \text{s.t. } & \boldsymbol{x}\in\mathcal{B}, \\ & \boldsymbol{x}\in\mathcal{Q}. \end{aligned} \tag{24}$$

Note that what is developed in this article can be easily extended to a polytope $\mathcal{P}$ and a Cartesian product of $|T|$ quadrics $\times_{t=1\ldots|T|}\mathcal{Q}(\Psi_t)$. This is discussed in Section 3.5.

In particular, we study two splitting algorithms: the Douglas-Rachford (DR) scheme, and the alternating projection method (AP). We consider three variants of the latter: one based on the exact projection and two based on the quasi-projection from Section 2.7, that approximates the projection via a geometric construction. Splitting algorithms exploit the separable structure of the problem, since the projection onto each of the sets that define the intersection, $\Omega := \mathcal{B}\bigcap\mathcal{Q}$, is easy to compute. They recently have been widely studied, and perform particularly well on certain classes of nonconvex problems.

A first convergence result for the (local) solution of the alternating projections in the nonconvex setting is presented in [12, Theorem 3.2.3] for sets that intersect transversally. This result is particularized to (nonempty and closed) semi-algebraic intersections in [13, Theorem 7.3] which we use in this work. A second result that is used in this paper is [21, Corollary 1], which is a convergence result for a (modified) Douglas-Rachford splitting. These two important propositions exploit the Kurdyka-Łojasiewicz properties of the indicator function of semi-algebraic sets, and are unfortunately local results: the theorem from [13] assumes that the starting point is *near* the intersection of the two considered sets, and the theorem from [21] proves the convergence to a *stationary point* of the problem of minimizing the distance to one of the sets, subject to being in the second set. We present an example where both methods fail to converge to a feasible point (Figs. 9 and 12), and propose a restart heuristic in Section 3.6.

The structure is the following: § 3.1 briefly recalls how to project onto a box, § 3.2 details the AP methods, and § 3.3 covers the DR splitting. A comparison table of all methods is presented in § 3.4, a power systems application is discussed in § 3.5, and a restart mechanism is given in § 3.6.

3.1 Projection onto a box

This projection is straightforward. Indeed given a point $\boldsymbol{x}^0$, it suffices to check for each dimension i whether this point violates the lower (respectively upper) bound and replace it accordingly. This gives Algorithm 4.

Note that for a more general polytope $\mathcal{P}$, the projection cannot be computed analytically. However the projection can be efficiently computed by solving the convex QP optimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{P}} \left\|\boldsymbol{x}^0-\boldsymbol{x}\right\|_2^2. \tag{25}$$

Algorithm 4 Projection onto the box (23)

```
Require: $\boldsymbol{x}^0 \in \mathbb{R}^n$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}^0$
  for $i = 1 \dots n$ do
    if $x_i^0 < \underline{x}_i$ then
      $x_i \leftarrow \underline{x}_i$
    else if $x_i^0 > \overline{x}_i$ then
      $x_i \leftarrow \overline{x}_i$
    end if
  end for
  return $\boldsymbol{x}$
```

3.2 Alternating projection method

The alternating projection method can be easily built by alternately projecting onto the quadric and onto the box. This gives Algorithm 5. Depending on whether we use the exact projection or one of the two quasi-projections detailed in § 2.7, we refer to the methods as follows: alternating projections with exact projection onto the quadric (APE), alternating projections with the centre-based quasi-projection (APC) or with the gradient-based quasi-projection (APG). $\mathrm{Pr}_{\mathcal{X}}$ stands for (one solution of) the projection onto a (non)convex set $\mathcal{X}$ and $\mathrm{P}_{\mathcal{Y}}$ the (quasi-)projection onto a set $\mathcal{Y}$.

Algorithm 5 Alternating projections

```
Require: $\boldsymbol{x}^0 \in \mathbb{R}^n$
  $k \leftarrow 0$
  while $k < n_{\text{iter}}$ and not $\boldsymbol{x}^k \in \Omega$ do
    $\boldsymbol{y}^{k+1} \leftarrow \mathrm{Pr}_{\mathcal{B}}(\boldsymbol{x}^k)$ {Using Algorithm 4}
    $\boldsymbol{x}^{k+1} \leftarrow \mathrm{P}_{\mathcal{Q}}(\boldsymbol{y}^{k+1})$ {Using Algorithm 2 or Algorithm 3}
    $k \leftarrow k + 1$
  end while
  return $\boldsymbol{x}^k$
```

Assuming that the initial iterate is close enough to the intersection, [13] provides a convergence result for APE, which is particularized to our case in Proposition 3.1. Note that Proposition 3.1 guarantees convergence to a point $\boldsymbol{x}^*$ in Ω, but provides no guarantee about the optimality of this point, *i.e.*, it is not true in general that $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\Omega} \left\|\boldsymbol{x} - \boldsymbol{x}^0\right\|_2^2$.

Proposition 3.1. *If Algorithm 5 with the exact projection (APE) is initialized from $\boldsymbol{x}^0 \in \mathcal{Q}$ and near $\mathcal{B}$, then the distance of the iterates to the intersection $\mathcal{Q} \bigcap \mathcal{B}$ converges to zero, and hence every limit point, $\boldsymbol{x}^*$, lies in $Q \bigcap \mathcal{B}$.*

Proof. This follows from [13, Theorem 7.3], since $\mathcal{Q}$ and $\mathcal{B}$ are semi-algebraic and $\mathcal{B}$ is bounded. □

Remark that if $\boldsymbol{B} \succ \boldsymbol{0}$, then $\mathcal{Q}$ is also bounded and we can as well choose $\boldsymbol{x}^0 \in \mathcal{B}$ and near $\mathcal{Q}$.

Figs. 10 and 11 present examples where the alternating methods converge in a single iteration or in multiple iterations. Only APC is depicted. Notice that if APE converges in a single iteration, then the obtained solution, $\boldsymbol{x}^*$, is an optimal solution of (24), that is, $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \Omega} \left\|\boldsymbol{x} - \boldsymbol{x}^0\right\|_2^2$. Figure 9 shows a pathological example where none of the alternating projection methods converge to a feasible point of (3). We propose in Section 3.6 certain heuristics in order to overcome such pathological cases.

3.3 Douglas-Rachford method

Following [21], the Douglas-Rachford splitting algorithm aims at solving

$$\min\ f(\boldsymbol{x}) + g(\boldsymbol{x})$$

where f has a Lipschitz continuous gradient and g is a proper closed function. The DR iteration starts at any $\boldsymbol{y}_0$ and repeats for $k = 0, 1, \ldots$

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \operatorname{prox}_f(\boldsymbol{y}^k), \\ \boldsymbol{y}^{k+1} &= \boldsymbol{y}^k + \operatorname{prox}_g(2\boldsymbol{x}^{k+1} - \boldsymbol{y}^k) - \boldsymbol{x}^{k+1}, \end{aligned}$$

where the prox operator (with step size 1) is defined as

$$\operatorname{prox}_f(v) = \arg\min_{x \in \mathbb{R}^n} \left(f(x) + \frac{1}{2}\|x - v\|_2^2 \right). \tag{26}$$

Let $\mathcal{I}_{\mathcal{X}} : \mathcal{X} \mapsto \mathbb{B}$ be the indicator function of a set $\mathcal{X}$ defined as

$$\mathcal{I}_{\mathcal{X}}(\boldsymbol{x}) = \begin{cases} 0 \text{ if } x \in \mathcal{X}, \\ +\infty \text{ else.} \end{cases}$$

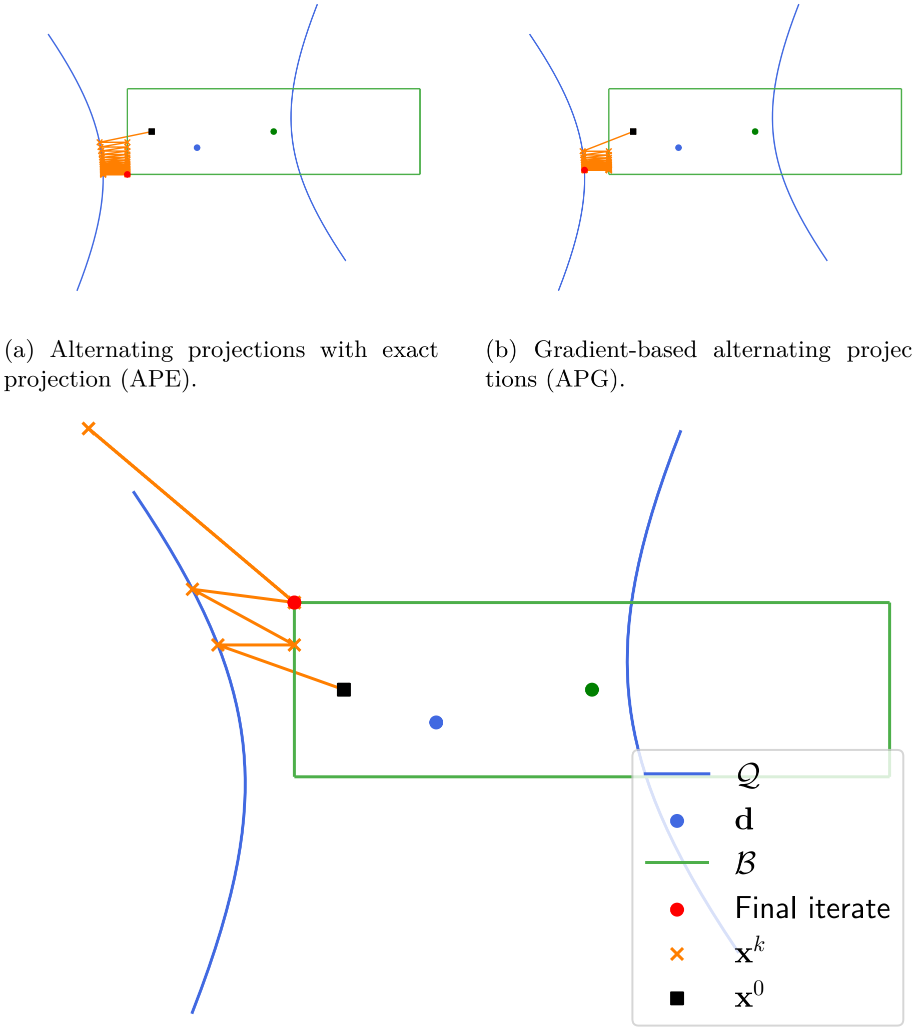
If we identify $f := \mathcal{I}_{\mathcal{B}}$ and $g := \mathcal{I}_{\mathcal{Q}}$, *i.e.*, the indicator functions of the sets that define Ω, then the DR algorithm reads

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \operatorname{Pr}_{\mathcal{B}}(\boldsymbol{y}^k) \\ \boldsymbol{y}^{k+1} &= \boldsymbol{y}^k + \operatorname{Pr}_{\mathcal{Q}}(2\boldsymbol{x}^{k+1} - \boldsymbol{y}^k) - \boldsymbol{x}^{k+1} \end{aligned}$$

which can be rewritten in a compact way [5],

$$\boldsymbol{x}^{k+1} = \left(\operatorname{Pr}_{\mathcal{Q}}(2\operatorname{Pr}_{\mathcal{B}} - \boldsymbol{I}) + (\boldsymbol{I} - \operatorname{Pr}_{\mathcal{B}})\right)(\boldsymbol{x}^k) \tag{27}$$

since the proximal operator of an indicator function of a given set X is the projection onto this set Pr_X. We denote this method as DR, and explicitly state it in Algorithm 6.

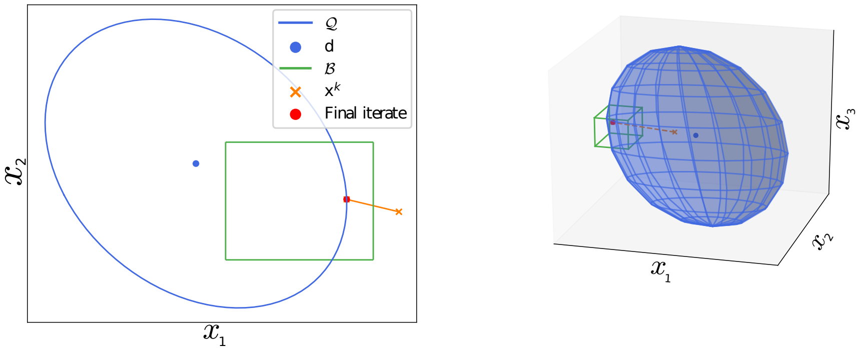


(a) Alternating projections with exact projection (APE).

(b) Gradient-based alternating projections (APG).

(c) Centre-based alternating projections (APC).

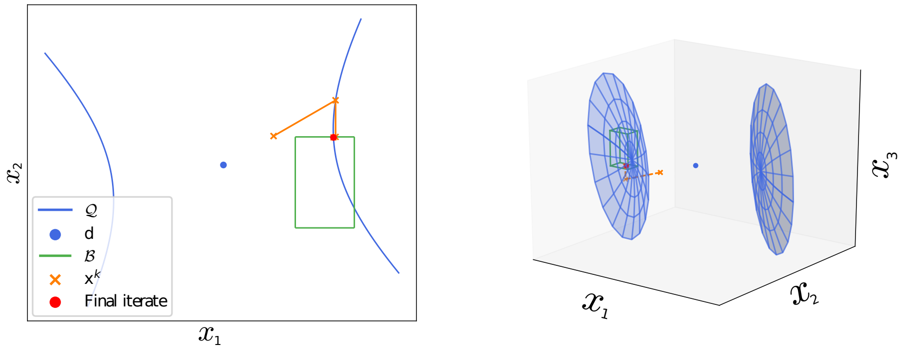
Figure 9: Illustration of a (2D) pathological case where none of the proposed alternating methods converge to a feasible point of (24). We represent both $\boldsymbol{x}^k$ and $\boldsymbol{y}^k$ as orange crosses.



(a) 2D (ellipse) case.

(b) 3D (ellipsoid) case.

Figure 10: Illustration of the centre-based alternating projection method (APC). In these cases, the method converges in a single iteration as the quasi-projection from Algorithm 3 yields a feasible point, *i.e.*, a point inside the box. We represent both $\boldsymbol{x}^k$ and $\boldsymbol{y}^k$ as orange crosses.



(a) 2D (hyperbola) case: the algorithm converges in three iterations.

(b) 3D (two-sheet hyperboloid) case, the algorithm converges in three iterations.

Figure 11: Illustration of successes of the centre-based alternating projection method (APC) on 2D and 3D hyperbolic cases. We represent both $\boldsymbol{x}^k$ and $\boldsymbol{y}^k$ as orange crosses.

Algorithm 6 Douglas-Rachford splitting method (DR)

```
Require: An initial point $\boldsymbol{x}^0$
  while a termination criterion is not met do
    $\boldsymbol{y}^{t+1} \leftarrow \mathrm{Pr}_{\mathcal{B}}(\boldsymbol{x}^t)$
    $\boldsymbol{z}^{t+1} \leftarrow \arg\min_{\boldsymbol{z}\in\mathcal{Q}} \left\|2\boldsymbol{y}^{t+1} - \boldsymbol{x}^t - \boldsymbol{z}\right\|^2$
    $\boldsymbol{x}^{t+1} \leftarrow \boldsymbol{x}^t + (\boldsymbol{z}^{t+1} - \boldsymbol{y}^{t+1})$
  end while
  return $\boldsymbol{z}^{t+1}$
```

Modified Douglas-Rachford We now present the modification of DR splitting for the feasibility problem of [21]. Instead of using the indicator function for the convex set $\mathcal{B}$, the splitting is performed with the squared distance function $d^2_{\mathcal{B}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}\in\mathcal{B}} ||\boldsymbol{x} - \boldsymbol{y}||_2^2$, *i.e.*,

$$\min_{\boldsymbol{x}\in\mathcal{Q}} d^2_{\mathcal{B}}(\boldsymbol{x})\,, \tag{28}$$

which can be equivalently seen as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} d^2_{\mathcal{B}}(\boldsymbol{x}) + \mathbb{1}_{\mathcal{Q}}(\boldsymbol{x})\,. \tag{29}$$

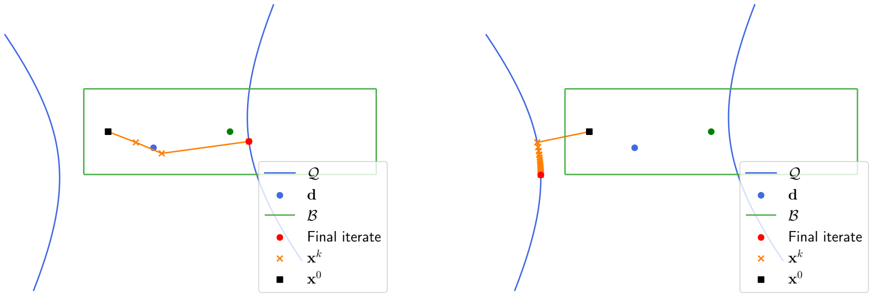
DR applied to (29) gives Algorithm 7, denoted as DR-F. We can use [21, Corollary 1] to obtain a convergence result for the DR-F method.

Algorithm 7 Douglas-Rachford splitting method for feasibility problems (DR-F)

```
Require: An initial point $\boldsymbol{x}^0$ and a step size parameter $\gamma > 0$
  while a termination criterion is not met do
    $\boldsymbol{y}^{t+1} \leftarrow \frac{1}{\gamma+1}(\boldsymbol{x}^t + \gamma P_{\mathcal{B}}(\boldsymbol{x}^t))$
    $\boldsymbol{z}^{t+1} \leftarrow \arg\min_{\boldsymbol{z}\in\mathcal{Q}} \left\|2\boldsymbol{y}^{t+1} - \boldsymbol{x}^t - \boldsymbol{z}\right\|^2$
    $\boldsymbol{x}^{t+1} \leftarrow \boldsymbol{x}^t + (\boldsymbol{z}^{t+1} - \boldsymbol{y}^{t+1})$
  end while
  return $\boldsymbol{z}^{t+1}$
```

Proposition 3.2. *If* $0 < \gamma < \sqrt{\frac{3}{2}} - 1$*, then the sequence* $\left\{(\boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{x}^t)\right\}$ *provided by Algorithm 7 converges to a point* $(\boldsymbol{y}^*, \boldsymbol{z}^*, \boldsymbol{x}^*)$ *which satisfies* $\boldsymbol{z}^* = \boldsymbol{y}^*$*, and* $\boldsymbol{z}^*$ *is a stationary point of (28).*

Proof. Since $\mathcal{Q}$ and $\mathcal{B}$ are nonempty closed semi-algebraic set, with $\mathcal{B}$ being convex and compact, we satisfy the hypothesis of [21, Corollary 1] for $0 < \gamma < \sqrt{\frac{3}{2}} - 1$. □



(a) Douglas-Rachford (DR).

(b) Modified Douglas-Rachford for the feasibility problem (DR-F).

Figure 12: Illustration of the behaviour of the DR splitting algorithms on the same (pathological) case of Figure 9. On this problem, DR does converge to a feasible point while DR-F does not. However, DR-F converges to a stationary point (a minimum) of (28).

Table 1: Comparison of the considered splitting methods.

	APE	APC	APG	DR	DR-F
Complexity	$\mathcal{O}(n^3 + kn^2)$	$\mathcal{O}(kn^2)$	$\mathcal{O}(kn^2)$	$\mathcal{O}(n^3 + kn^2)$	$\mathcal{O}(n^3 + kn^2)$
Convergence guarantees	Locally to a feasible point of (24): Proposition 3.1	None	None	None[1]	Locally to a stationary point of (28): Proposition 3.2

3.4 Comparison

Table 1 compares the different complexities and convergence results of all the methods. Methods using exact projection onto the quadric require the diagonalization of $\boldsymbol{B}$ as a precomputation step, which typically costs $\mathcal{O}(n^3)$ flops.

3.5 Extensions and applications

We can extend the splitting methods to a polytope, $\mathcal{P}$, and a Cartesian product of m quadrics $\mathcal{Q}^{\text{tot}} := \times_{i=0}^{m} \mathcal{Q}_i$, and solve

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \; & \left|\left|\boldsymbol{x} - \boldsymbol{x}^0\right|\right|_2^2 \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{P}, \\ & \boldsymbol{x} \in \mathcal{Q}^{\text{tot}}. \end{aligned} \tag{30}$$

The extension of all methods described in Table 1 is direct: instead of computing the projection on $\mathcal{B}$—now $\mathcal{P}$—analytically, we have to resort

[1] There are, however, proofs of the convergence of DR in some nonconvex applications, see, *e.g.*, the discussion in [2, Section 4].

to a QP solver. And, similarly to the retraction from [29], the (quasi-)projection is obtained by working independently on each quadric $\mathcal{Q}_i$:

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{Q}^{\text{tot}}} \left\|\boldsymbol{x}-\boldsymbol{x}^0\right\|_2 \Leftrightarrow \boldsymbol{x}_i^* \in \arg\min_{\boldsymbol{x}_i\in\mathcal{Q}_i} \left\|\boldsymbol{x}_i-\boldsymbol{x}_i^0\right\|_2 \; \forall i = 1\ldots m. \quad (31)$$

For example, in the practical case from [29], the paper focuses on the dynamic economic dispatch problem which aims at the optimal allocation of power production among generating units at each timestep, *e.g.*, each hour of a day. The modelling of the power losses makes the feasible set of each independent (static) economic dispatch a quadric and certain operational constraints, namely the ramping constraints, couple consecutive time steps. Hence, the full feasible set Ω is a polytope $\mathcal{P} \subseteq \mathbb{R}^{nT}$ that accounts for the power ranges (box) and ramping constraints, and of a Cartesian product of T different quadrics $\mathcal{Q}_i \subseteq \mathbb{R}^n$ that model the balance constraint, *i.e.*, that power production matches demand. The projection of a point onto $\Omega = \mathcal{P} \bigcap \mathcal{Q}^{\text{tot}}$ can then be obtained using the methods described in the present paper.

Moreover, the point that has to be projected in [29] is obtained as the solution of a surrogate problem defined on a relaxed set, see [30, 28, 29] for more details. And because this relaxation is close to the feasible set, the point that has to be projected is inside the box and near the quadric. This is the reason for the favorable performance of APC which is reported in Section 4.2.

3.6 Implementation details

To address the convergence issues identified in Figs. 9 and 12, we add a restart mechanism whenever this situation arises. Such situations are easily detected: the alternating method will loop between two points, and the DR or DR-F will simply converge to an infeasible point. These problems mostly appear in the hyperboloid case, and typically occur when the method is trapped on the wrong sheet of the hyperboloid. To mitigate this issue, when detected, we use the geometric construction from the centre-based quasi-projection (Algorithm 3 with $\boldsymbol{\xi} = \boldsymbol{x}^0 - \boldsymbol{d}$), and select the *largest* β. This is equivalent to transforming $\boldsymbol{x}^k \in \mathcal{Q}, \boldsymbol{x}^k \notin \mathcal{B}$ into $-\boldsymbol{x}^k =: \boldsymbol{x}^{k+1} \in \mathcal{Q}$, and continuing the method from $\boldsymbol{x}^{k+1}$. Alternatively, it is also possible to consider $\boldsymbol{x}^{k+1}$ such that at least one—instead of all—of its components is the opposite of $\boldsymbol{x}^k$. If, on the other hand, $\boldsymbol{x}^k \in \mathcal{B}$, then we work analogously with respect to the centre of the box.

Such a restart mechanism is not a guarantee of convergence: the method can then be trapped into another region, or even come back to the exact same region. But in the few instances ($\approx$ once every 10000 trials) where the presented algorithms experience convergence issues, the restart results in successful convergence.

4 Numerical experiments

This section is devoted to the benchmarking of the methods developed in Section 3.

Section 4.1 tests the five presented methods (APE, APC, APG, DR, DR-F) as well as IPOPT. IPOPT is an interesting method to benchmark against, as it is a natural candidate for solving (24). Note that IPOPT is an open-source solver that uses an embedded linear solver. The performance of IPOPT can be enhanced through the use of a dedicated commercial linear solver. In this work, we use Pardiso [1].

We solve for small scale (Figs. 13 and 15) and larger scale (Figs. 14 and 16) instances of (24). For each considered dimension, n, we run 100 randomly independent trials in order to smooth the effect of the random selection of the problem parameters. In particular, each independent trial consists of a unique (randomly generated) set of parameters $\boldsymbol{B}, \boldsymbol{b}, c$ and $\boldsymbol{x}^0 \in \mathbb{R}^n$.

The ellipsoid case is tested in Section 4.1.1 and the hyperboloid case in Section 4.1.2.

The problems are generated as follows: we first create a quadric with $\boldsymbol{A} \sim \mathcal{N}(\boldsymbol{\mu} = \mathbb{1}, \Sigma = I)$, $\boldsymbol{B} = \frac{\boldsymbol{A}+\boldsymbol{A}^\intercal}{2}$, $\boldsymbol{b}$ defined with $b_i \sim \mathcal{N}(\mu = 0, \sigma = 1)$ and $c \sim \mathcal{N}(\mu = -1, \sigma = 1)$. For the ellipsoidal case, we shift $\boldsymbol{B}$ in order to ensure that $\boldsymbol{B} \succ \boldsymbol{0}$. Then, we find one feasible point and construct the box $\mathcal{B}$ around it; this allows us to ensure that the intersection of $\mathcal{B}$ and $\mathcal{Q}$ is nonempty. Note that this feasible point is not necessarily the centre of the box.

Then, in Section 4.2, we perform two experiments for comparing APC to Gurobi for a very specific problem structure, which is a problem stemming from [29], and the initial goal of the present research. The same remarks with respect to the 100 randomly generated data also apply here.

For this second experiment, we use Gurobi as a benchmark because i) it may also be a natural method for solving (24)[2] and ii) it provides lower bounds, which allow us to assess how close the returned solutions are to the global optimum. We benchmark it against APC because, even if it is the worst-performing methods among the five that are presented in Section 4.1, it still outperforms Gurobi. This behaviour is explained by the relative position of the starting point with respect to the feasible set from the problem of Section 4.2: this point is inside the box and close to the quadric.

Note that, in all experiments, whenever an algorithm terminates with a timeout and returns an infeasible point, the associated objective is meaningless. In order to avoid distorting our reported results, we omit these instance in the recorded objectives; but we count the number of timeouts and record the deviation.

The deviation is computed as

$$\text{deviation} = \left| \boldsymbol{x}^\intercal \boldsymbol{B} \boldsymbol{x} + \boldsymbol{b}^\intercal \boldsymbol{x} + c \right| , \tag{32}$$

and is an intuitive measure of how far an infeasible point is to the feasible set. The prescribed tolerance for the deviation is 10^{-6}. This deviation does not account for the box. This is not an issue here, because none of the tests considered in the numerical experiments terminates outside of the box.

[2] Since version 9.0, Gurobi supports nonconvex QCQP optimization, and Gurobi is employed widely in the power systems optimization community.

4.1 Douglas-Rachford, Alternating Projections and IPOPT

Two different settings are considered here. In Section 4.1.1, the matrix $\boldsymbol{B}$ is chosen such that $\boldsymbol{B} \succ \boldsymbol{0}$, *i.e.*, the quadric is an *ellipsoid*. This means that the quasi-projection with $\boldsymbol{\xi} = \boldsymbol{x}^0 - \boldsymbol{d}$ is well-defined: situations depicted in Figure 8 cannot occur. In Section 4.1.2, we consider the case of *hyperboloids*, *i.e.*, $\boldsymbol{B}$ is nonsingular but indefinite.

From these two experiments, it appears that both DR-F and APE are the methods that find the best solution in terms of objective. However, if the execution time is taken into account, APG reaches an objective close to the one of DR-F and APE in a significantly lower run time. APG should therefore be considered, *e.g.*, if the eigenvector decomposition is too expensive to compute. APC works particularly well in the ellipsoidal case, but performs worse in the hyperboloidal case. IPOPT is clearly the slowest method. It achieves good solution objectives in the ellipsoidal case, but gives poorer results in the hyperboloidal case.

4.1.1 Ellipsoid experiments

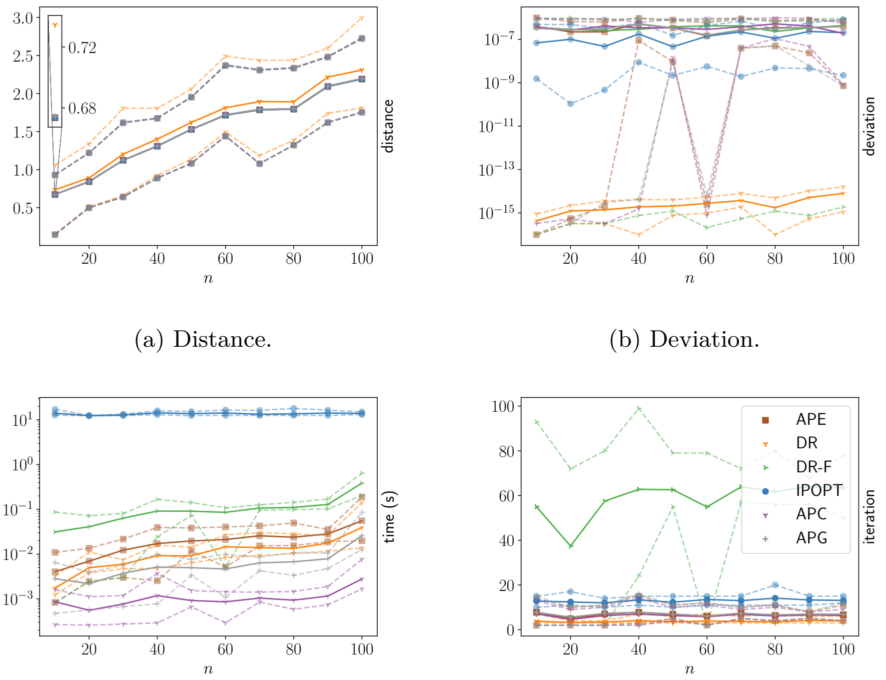
In these two experiments, we run small and large-scale ellipsoidal problems. The box $\mathcal{B}$ is small with respect to the quadric and the starting points $\boldsymbol{x}^0$ are uniformly distributed inside the box.

For small-scale ellipsoidal problems ($n \leqslant 100$, Figure 13), we observe that all methods except DR reach the same objective: APE, DR-F and IPOPT obtain the same objective, APG is within 1% and DR within several percent. We also observe that none of the methods exceeds the prescribed deviation accuracy of 10^{-6}, and that IPOPT provides the most feasible points.

The number of iterations required for each method remains more or less constant when the dimension increases. Considering the running time, APC is the fastest and ten times faster than APG, which is two time faster than DR. DR-F and APE require approximately the same amount of time, which is two times slower than DR. Finally, IPOPT requires much more time than all the other methods.

For large-scale ellipsoidal problems ($n \geqslant 100$, Figure 14), the behaviour of the methods remains similar as the small-scale case. The distance increase with n is simply due to the increase of $\left\|\boldsymbol{x}^* - \boldsymbol{x}^0\right\|_2$ with n.

Remark that the execution time of IPOPT is remarkably stable, this is because creating the model already requires approximately 10 seconds, and this creation time does not increase much when the dimension increases. However, it should be noted that i) for much larger dimension $n >> 1000$ the solving time of IPOPT increases significantly and ii) for such large dimension, it becomes crucial to use advanced linear algebra tools for, *e.g.*, the eigenvalue decompositions and matrix products used in the methods developed here. Hence, the comparison against IPOPT when the latter relies on a dedicated linear algebra software (Pardiso) becomes less meaningful for too large n.



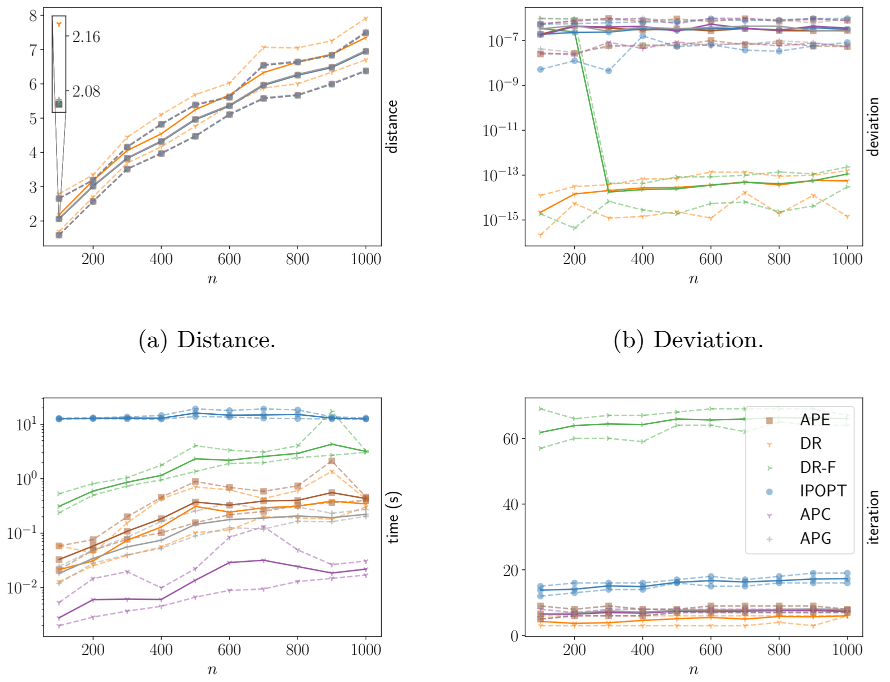
(a) Distance.

(b) Deviation.

(c) Execution time in seconds.

(d) Number of iterations.

Figure 13: Comparison of the different methods developed in Section 3: Douglas-Rachford splitting (DR) and its modified counterpart (DR-F), alternating projections using the exact projection (APE) and the alternating projections using the quasi-projections (centre-based APC and gradient-based APG). IPOPT is used as a benchmark with standard settings and with the underlying linear solver Pardiso. Ten dimensions n are considered and, for each n, 100 independent trials with $\boldsymbol{B} \succ \boldsymbol{0}$ are run. The top (bottom) dashed lines represent the max (min) value of the 100 trials, and the continuous line is the sample mean. The frame in the upper left of the upper left panel is a magnification around $n = 10$.



(a) Distance.

(b) Deviation.

(c) Execution time in seconds.

(d) Number of iterations.

Figure 14: Same as Figure 13 for larger dimensions.

4.1.2 Hyperboloid experiments

In these two experiments, we run small and large-scale hyperboloidal problems. The box $\mathcal{B}$ is large with respect to the quadric and the starting point $\boldsymbol{x}^0$ is uniformly distributed inside $\mathcal{B}$.

For small-scale hyperboloidal problems ($n \leqslant 100$, Figure 15), we observe that the best objectives are obtained by APE. Then, within several %, by DR-F, APG, DR and IPOPT. These four methods reach the same solution as APE most of the time, however they sometimes reach solutions that are far away from the best methods, see, *e.g.*, the maximum curve (top dashed-lines in Figure 15a) that is significantly above the maximum curves of APE. Finally, APC performs poorly in terms of objective values. It is now APG which is the fastest method, despite its need of more iterations: the reason stems from the need of APC to resort to an exact projection whenever the situation depicted in Figure 8 appears.

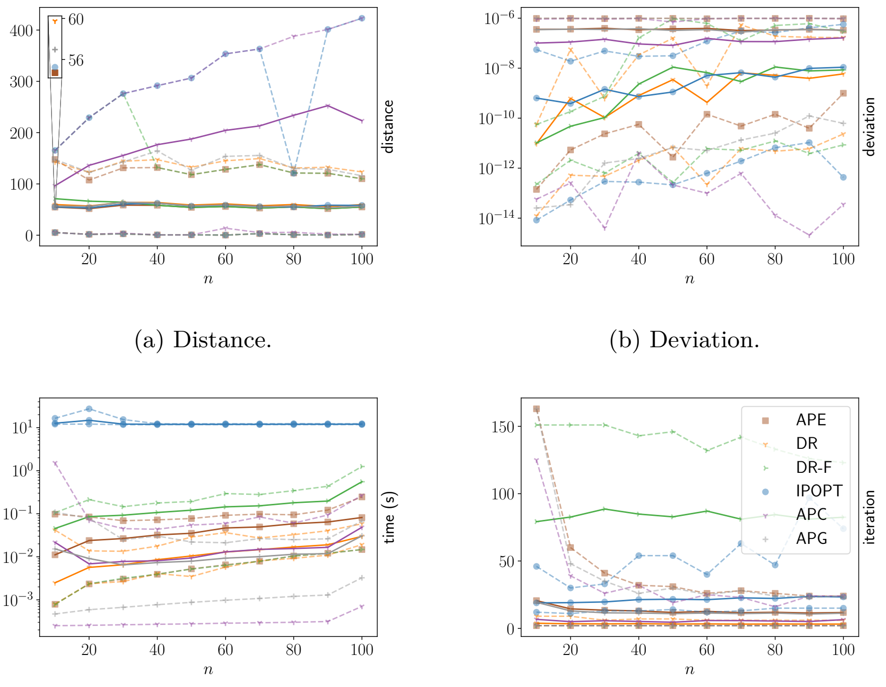
For the large-scale hyperboloidal problems ($n \geqslant 100$, Figure 16), the best objectives are attained by APE and DR-F. The APG algorithm comes within one percent of their performance. The unmodified Douglas-Rachford finds objectives within several percent, and IPOPT within 10 percent, *e.g.*, the mean objective for $n = 1000$ (solid lines in Figure 16a) is around 0.51 for APE, DR-F and APG, around 0.53 for DR and around 0.63 for IPOPT. We note that the number of iterations increases with n, and that APG is also the fastest method. We also observe a significant increase in the execution time of IPOPT, which implies that the solving time is now larger than the 10 seconds that are required for creating the problem. Finally, we observe that both IPOPT and APC sometimes finish with a timeout, and return points above the prescribed deviation of 10^{-6}.

4.2 Alternating projections versus Gurobi

In this section, we benchmark the alternating projections with centre-based quasi-projection (APC) against an *exact method*, which actually aims at finding the optimal solution of (24). Several methods can be used to tackle (24), here we choose to use the commercial software Gurobi [17]. This tool tackles the nonconvex quadratic equality via piecewise linearization, and solves the resulting mixed-integer quadratic programming (MIQP) problem. In this way, a solution along with a lower bound are obtained, and the optimal solution—up to a given tolerance—can be reached, assuming enough time is afforded to the solver.

The problem parameters are chosen so as to resemble problems from the power systems literature: the feasible set of the economic dispatch problem with power losses [29] typically exhibits a similar structure as the feasible set of (24). The entries of $\boldsymbol{B}$ are in the order of 10^{-5} except for diagonal entries (10^{-4}), $\boldsymbol{b}$ is close to 1 and c around -100. $\boldsymbol{B}$ represents the quadratic power losses—expected to be small—, and $\boldsymbol{b}$ encodes the constraint stating that the sum of the power production must be equal to the demand ($\approx -c$).

The following quantities are compared:



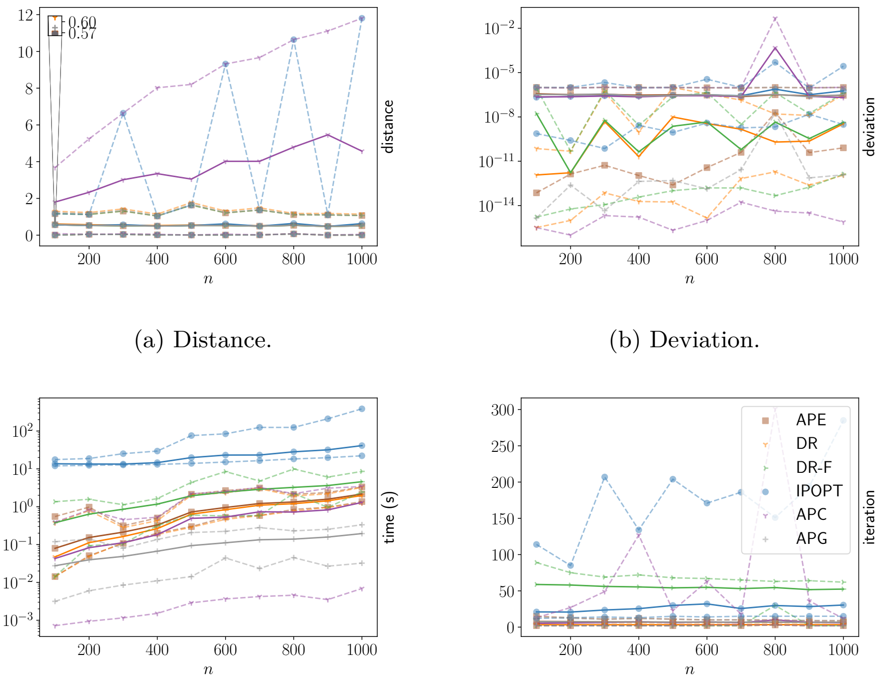
(a) Distance.

(b) Deviation.

(c) Execution time in seconds.

(d) Number of iterations.

Figure 15: Same as Figure 13 with $\boldsymbol{B} \nsucceq \mathbf{0}$.



(a) Distance.

(b) Deviation.

(c) Execution time in seconds.

(d) Number of iterations.

Figure 16: Same as Figure 15 for larger dimensions.

$$\text{Relative time} = \log\left(1 + \frac{t_{\text{APC}} - t_{\text{Gur}}}{t_{\text{Gur}}}\right) = \log\left(\frac{t_{\text{APC}}}{t_{\text{Gur}}}\right)$$
$$\text{Relative distance} = \log\left(\frac{d_{\text{APC}}}{d_{\text{Gur}}}\right)$$

where t_{APC}, t_{Gur} are the execution times of APC and Gurobi, respectively, and d is the distance between the final iterate and $\boldsymbol{x}^0$, *i.e.*, the objective.

Note that, in order to smooth out random effects, we run $m = 100$ slightly different instances for each dimension and report the mean, such that, *e.g.*,

$$t_{\text{APC}} = \frac{\sum_{i=1}^{m} t_{\text{APC}}^{i}}{m},$$

where t_{APC}^{i} stands for the execution time of the $i-$th instance of the alternating projection method.

In the following experiments, it may be the case that no feasible point is found. For the alternating projection method, this can occur when the method reaches the maximum number of iterations, *e.g.*, when the method is trapped in a cycle loop (see Figure 9). On the other hand, Gurobi may also fail to yield a feasible point, if the time limit criterion is attained. We do not encode such points into the relative distance. In this way, we do not pollute the reported distance mean by a small number of instances that terminate due to a timeout. However, we also record the number of timeouts—either due to maximum iterations or time limit.

4.2.1 One-shot experiment

In this experiment, we aim to compare the speed of both methods. We thus terminate the algorithm as soon as it finds a feasible point, hence the reference to "one shot". For APC, this does not affect the algorithm. On the other hand, Gurobi relies on lower and upper bounds, and terminates whenever a targeted tolerance is achieved. Here, we modify the stopping criterion such that the algorithm stops as soon as a feasible solution is obtained, no matter the objective. Hence, this is a lower bound on the execution time if the method is run with the tolerance criterion.

Figure 17 presents the relative execution time and distance. We observe that APC clearly outperforms Gurobi in this experiment: for low dimension APC executes at least two times faster and reaches a better solution, for larger dimensions the difference becomes even larger, *e.g.*, when the dimension is bigger than 40 APC accelerates by a factor of 100 000 and reaches an objective which is 10 times lower than that of Gurobi. Moreover, it should be noted that the number of timeout terminations recorded in Gurobi starts to increase for dimensions greater than 40 (see the bar plot in Figure 17): hence the relative time is limited because of the time limit, this explains the saturation of the relative time for large dimensions. The relative distance does not encode the infeasibility of the points that finish with a timeout, and such points should have an infinite objective value. The time limit criterion is set to 600 seconds. We note that a significant number of instances terminate without a solution for problems of large dimension.

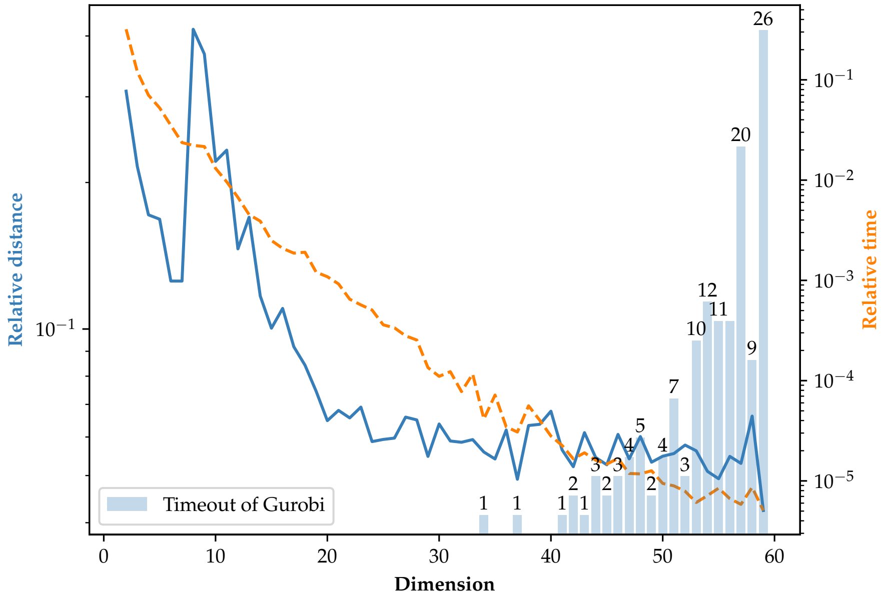


Figure 17: Comparison between the alternating projection method with the centre-based quasi-projection (APC) and Gurobi. In this experiment, described in Section 4.2.1, the method terminates whenever it finds a feasible solution, no matter the objective. The timeout termination of Gurobi is set at 600 seconds, and the number of timeouts for the 100 instances is depicted as a bar plot.

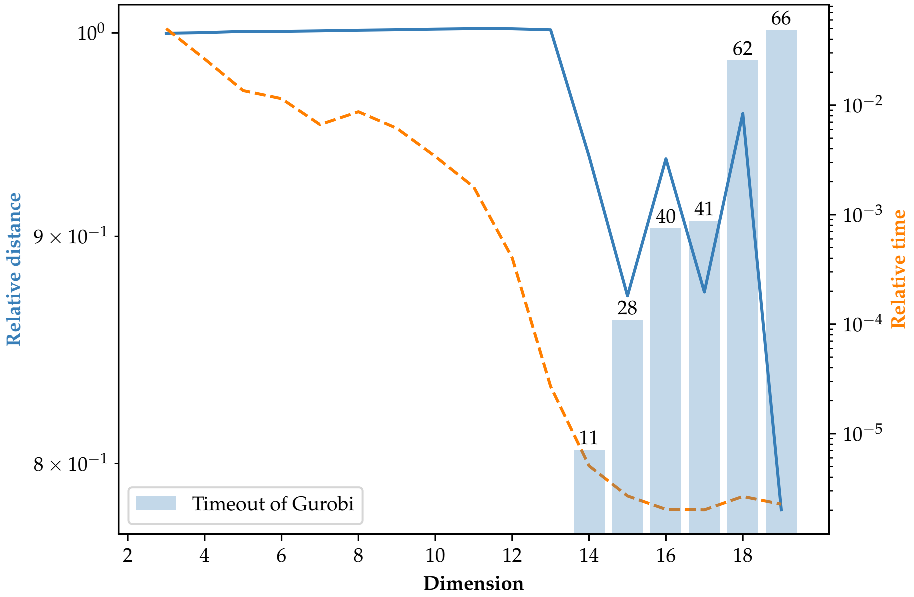


Figure 18: Comparison between the alternating projection method with the centre-based quasi-projection (APC) and Gurobi. In this experiment, described in Section 4.2.2, Gurobi terminates when the objective is proven to be optimal within a 1% tolerance. The timeout termination of Gurobi is set to 600 seconds.

4.2.2 Multiple-shot experiment

In this experiment, we allow Gurobi to execute until it reaches the best solution, up to a given tolerance of one percent, or until timeout (600 seconds). Figure 18 depicts the (mean) relative distance and execution time, for 100 runs, as well as the number of timeout terminations of Gurobi. We observe that, for small problem instances, *i.e.*, when the dimension is below 13, Gurobi reaches the best solution *which is also very close to the one obtained via APC*. Indeed, a relative distance around one means that the solutions returned by the two algorithms are comparable. Since Gurobi does not terminate with a timeout, this implies that the solution returned by APC is, as a matter of fact, also optimal. Note that the theory does not guarantee that this should occur. We also note that the execution time of Gurobi is 10 to 1000 larger than that of APC. For higher dimension, the relative distance slightly decreases and the relative time converges to 1×10^{-6}: this is due to the increasing number of timeout terminations. In other words, Gurobi fails to find the best solution in an increasing execution time.

5 Conclusion

In this paper, the projection onto quadratic hypersurfaces, or quadrics, is investigated. We assume that the quadratic hypersurface is a non-cylindrical central quadric; however, the non-cylindrical assumption can be easily lifted by focusing on the variables that appear in the normal form. Using the method of Lagrange multipliers, we reduce this nonconvex optimization problem to the problem of finding the solutions of a system of nonlinear equations. We then show how one of the optimal solutions of the projection either lies in a finite set of computable solutions, or is a root of a scalar-valued nonlinear function. This unique root is located on a given interval, and is therefore simply computed with a Newton-Raphson scheme to which we provide suitable starting points. The cost of this projection is thus cheap, and the bottleneck is the eigenvector decomposition. This decomposition is needed for diagonalizing the matrix that is used to define the quadric.

We also propose a heuristic, referred to as quasi-projection, based on a geometric construction. This construction consists of finding the closest intersection between the quadric and a line passing by the point that we want to project. We detail two variants of the quasi-projection, depending on whether the direction of the line is computed as the level-curve gradient of the quadric, or the vector joining the centre and the point. This quasi-projection does not require eigenvector decomposition, thereby economizing in computational time.

This projection is then leveraged in the context of splitting algorithms, namely alternating projections and Douglas-Rachford splitting. This allows us to project a point onto a feasible set that is the intersection between a quadric and a box. The extension to the more general case of a Cartesian product of quadrics and a polytope is also discussed. Five methods are proposed depending on whether we use standard Douglas-Rachford splitting (DR), modified Douglas-Rachford splitting (DR-F), or one of the alternating projection methods. We detail the alternating projections with the exact projection on the quadric (APE) or one of the two quasi-projections (the centre-based APC or the gradient-based APG).

All methods are tested on problems of several dimensions, from 10 to 1000, and 100 independent trials are executed for each dimension. Using IPOPT as a benchmark, we find that APE and DF reach the best objectives and APG is within one percent, while APC and IPOPT lag behind. However, APG is much faster than the other methods and appears to achieve a good trade-off between the attained objective and execution time.

We also test APC on a case similar to the economic dispatch problem from the power systems literature, and compare it to Gurobi. We show that, in this specific case where the initial point is close to the feasible set, APC quickly reaches a solution close to or better than Gurobi, even if the execution time of Gurobi is several orders of magnitude greater. For small dimensions, Gurobi can guarantee the optimality of its solution, which shows that APC obtains the optimal solution in our examples. For higher dimension, Gurobi terminates with a timeout and with a higher objective than APC. Hence, even APC, which is the poorest of the methods that we propose in our paper in terms of performance, outperforms Gurobi in these experiments.

For the first part of the paper, namely the projection onto nonsingular quadrics in Section 2, the extension to singular quadrics could be contemplated. In this context, the linear independence constraint qualification, LICQ, is not fulfilled any more. We should also include such points as projection candidates. A numerical comparison with the method from [27] is another natural research direction for further work.

For the second part of the paper, Section 3, further research may include the comparison with the alternating projection method using inexact projections, *e.g.*, projecting on the tangent space of the previous feasible point. Such methods are discussed in [14].

Acknowledgement

This work was supported by the Fonds de la Recherche Scientifique - FNRS under Grant no. PDR T.0025.18.

References

[1] Christie Alappat, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. A recursive algebraic coloring technique for hardware-efficient symmetric sparse matrix-vector multiplication. *ACM Trans. Parallel Comput.*, 7(3), June 2020.

[2] Francisco J. Aragón Artacho, Rubén Campoy, and Matthew K. Tam. The Douglas–Rachford algorithm for convex and nonconvex feasibility problems. *Mathematical Methods of Operations Research*, 91(2):201–240, April 2020.

[3] Tilo Arens, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger, and Hellmuth Stachel. *Mathematik*. Spektrum Akademischer Verlag, 1. aufl. 2008 edition, 2 2008.

[4] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on the Kurdyka-Łojasiewicz Inequality. *Mathematics of Operations Research*, 35(2):438–457, May 2010.

[5] Heinz H. Bauschke, Patrick L. Combettes, and D. Russell Luke. Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. *Journal of the Optical Society of America A*, 19(7):1334, July 2002.

[6] Amir Beck. *Introduction to Nonlinear Optimization.* MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, October 2014.

[7] Pierre B. Borckmans, S. Easter Selvan, Nicolas Boumal, and P.-A. Absil. A Riemannian subgradient algorithm for economic dispatch with valve-point effect. *J. Comput. Applied. Math.*, 255:848–866, 2013.

[8] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, UK ; New York, 2004.

[9] Scott Brown. *Local Model Feature Transformations.* PhD thesis, The University of South Alabama, may 2020.

[10] Richard L. Burden and J. Douglas Faires. *Numerical analysis.* Brooks/Cole, 7th edition, 2001.

[11] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods.* MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, January 2000.

[12] D. Drusvyatskiy. *Slope And Geometry In Variational Mathematics.* PhD thesis, Cornell university, August 2013.

[13] D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. Transversality and Alternating Projections for Nonconvex Sets. *Foundations of Computational Mathematics*, 15(6):1637–1651, December 2015.

[14] D. Drusvyatskiy and A. S. Lewis. Local Linear Convergence for Inexact Alternating Projections on Nonconvex Sets. *Vietnam Journal of Mathematics*, 47(3):669–681, September 2019.

[15] James Fletcher and Warren B. Moors. Chebyshev sets. *Journal of the Australian Mathematical Society*, 98(2):161–231, April 2015.

[16] Gene H. Golub and Charles F. Van Loan. *Matrix computations.* Johns Hopkins studies in the mathematical sciences. The Johns Hopkins University Press, Baltimore, fourth edition edition, 2013.

[17] Gurobi Optimization Inc. Gurobi Optimizer Reference Manual, 2018.

[18] Shih-Feng Huang, Yung-Hsuan Wen, Chi-Hsiang Chu, and Chien-Chin Hsu. A Shape Approximation for Medical Imaging Data. *Sensors*, 20(20):5879, January 2020.

[19] A. S. Lewis, D. R. Luke, and Jérôme Malick. Local Linear Convergence for Alternating and Averaged Nonconvex Projections. *Foundations of Computational Mathematics*, 9(4):485–513, August 2009.

[20] A. S. Lewis and Jérôme Malick. Alternating Projections on Manifolds. *Mathematics of Operations Research*, 33(1):216–234, February 2008.

[21] Guoyin Li and Ting Kei Pong. Douglas–Rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Mathematical Programming*, 159(1):371–401, September 2016.

[22] Gus K. Lott III. Direct Orthogonal Distance to Quadratic Surfaces in 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1888–1892, September 2014.

[23] D. Martínez Morera and J. Estrada Sarlabous. On the distance from a point to a quadric surface. *Investigación Operacional*, 24(2):153–161, September 2013.

[24] Yurii Nesterov. *Lectures on Convex Optimization.* Springer Optimization and Its Applications. Springer International Publishing, 2 edition, 2018.

[25] Boris Odehnal, Hellmuth Stachel, and Georg Glaeser. *The Universe of Quadrics.* Springer-Verlag, Berlin Heidelberg, 2020.

[26] Chenhui Song, Jun Xiao, Guoqiang Zu, Ziyuan Hao, and Xinsong Zhang. Security region of natural gas pipeline network system: Concept, method and application. *Energy*, 217:119283, February 2021.

[27] Wilfredo Sosa and Fernanda MP Raupp. An algorithm for projecting a point onto a level set of a quadratic function. *Optimization*, pages 1–19, October 2020.

[28] Loïc Van Hoorebeeck, P.-A. Absil, and Anthony Papavasiliou. Global solution of economic dispatch with valve point effects and transmission constraints. *Electric Power Systems Research*, 189:106786, 2020.

[29] Loïc Van Hoorebeeck, P. A. Absil, and Anthony Papavasiliou. Solving non-convex economic dispatch with valve-point effects and losses with guaranteed accuracy. *International Journal of Electrical Power & Energy Systems*, 134:107143, January 2022.

[30] Loïc Van Hoorebeeck, Anthony Papavasiliou, and P.-A. Absil. MILP-based algorithm for the global solution of dynamic economic dispatch problems with valve-point effects. *IEEE Power and Energy Society General Meeting*, 2019.

[31] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, May 2006.