A speed up strategy for gradient methods

Anna De Magistris[1,2*], Serena Crisci[1,2], Valentina De Simone[1,2], Gerardo Toraldo[1,2]

[1]Department of Mathematics and Physics, University of Campania "Luigi Vanvitelli", Viale Lincoln 5, Caserta, 81100, CE, Italy.
[2]The authors are members of the INdAM research group GNCS, which partially supported this work.

*Corresponding author(s). E-mail(s): anna.demagistris@unicampania.it;
Contributing authors: serena.crisci@unicampania.it;
valentina.desimone@unicampania.it; gerardo.toraldo@unicampania.it;

Abstract

In this paper, we propose a new acceleration strategy for gradient-based methods applied to strictly convex Quadratic Programming (QP) problems. The strategy consists in performing, at selected iterations, minimization steps along alternative descent directions or even within low-dimensional affine subspaces. In particular, considering the contribution of the linear and quadratic part of the objective function could be useful in designing line searches in acceleration steps. We present numerical experiments to assess the impact of acceleration steps on the performance of different gradient methods. We examined randomly generated QP and box constrained QP test problems, designed to assess the algorithms under various conditions, such as matrix dimensions, condition numbers, and initialization strategies. Our experiments show that the use of acceleration steps in some Barzilai-Borwein methods significantly improves computational results. Moving to general minimization problems, the extension of our approach is not straightforward; in particular, it is not possible to directly extend the two-dimensional minimization phase. In this work, we take a first step in this direction by providing preliminary ideas for a possible extension of the accelerated algorithm to the minimization of general functions.

Keywords: gradient methods, convex quadratic programming, acceleration strategies, bound constrained quadratic programming

1 Introduction

We are primarily interested in designing convergence acceleration techniques for gradient-based methods to solve the problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \equiv \tfrac{1}{2}\,\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top \mathbf{x}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $\mathbf{b} \in \mathbb{R}^n$. This problem, possibly adding simple bounds or linear constraints on the variables, is of practical importance in several applications (see e.g. [1–6]).

Gradient methods still occupy a prominent place in the field of optimization methods thanks to their low computational cost and reduced storage requirements. Furthermore, problem (1) is a useful framework for testing algorithmic strategies in gradient methods that can subsequently be applied to more general problems. Indeed, in this paper we do not restrict our attention solely to the convex quadratic case, showing a possible way to extend our ideas to a more general context.

Starting from $\mathbf{x}_0 \in \mathbb{R}^n$, gradient methods generate a sequence $\{\mathbf{x}_k\}$ as follows

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k,$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ and $\alpha_k > 0$ is the steplength. In particular, in the classical Cauchy Steepest Descent (SD) method, where the steplength selection is based on the exact minimization rule, we have

$$\alpha_k = \frac{\mathbf{g}_k^\top \mathbf{g}_k}{\mathbf{g}_k^\top A \mathbf{g}_k}, \tag{2}$$

which guaranties the Q-linear rate of convergence

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|_A \le \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \|\mathbf{x}_k - \mathbf{x}^*\|_A,$$

where $\lambda_{\max}, \lambda_{\min}$ are the respectively the maximum and the minimum eigenvalues of A and $\|\mathbf{v}\|_A = \mathbf{v}^\top A\mathbf{v}$ for any $\mathbf{v} \in \mathbb{R}^r$. It is well known that the Cauchy algorithm performs very badly, even for mild conditioned problems, since the worst case predicted by (1) is likely to happen.

This has been explained by Akaike [7], who proved that the sequence generated by (1) gets trapped in the two-dimensional space spanned by the eigenvectors associated to $\lambda_{\max}$ and $\lambda_{\min}$, very slowly if the ratio $\lambda_{\max}/\lambda_{\min}$ is large, without being able to eliminate from the basis of the current search direction any of the two eigencomponents and hence to align the gradient with an eigendirection of the Hessian matrix. To overcome this drawback, many rules have been proposed [8–10], mainly related to the Barzilai-Borwein (BB) formulas

$$\alpha_k^{\mathrm{BB1}} = \frac{\|\mathbf{s}_{k-1}\|^2}{\mathbf{s}_{k-1}^\top \mathbf{y}_{k-1}},\ \alpha_k^{\mathrm{BB2}} = \frac{\mathbf{s}_{k-1}^\top \mathbf{y}_{k-1}}{\|\mathbf{y}_{k-1}\|^2},$$
$$\text{where } \mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1},\ \mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1},$$

proposed by Barzilai and Borwein in their seminal work [11] in 1988. The nice behavior of the BB methods is often explained with the nonmonotonicity of such algorithms producing an erratic path of the steplengths in the interior of the spectrum of A^{-1} which fosters not getting caught into badlands [12]. Since then, several gradient methods have been proposed moving from BB rules: gradient methods with retards, cyclic, adaptive BB-inspired rules [9, 10, 13, 14].

In particular, in [10] the authors proposed the Adaptive Barzilai-Borwein (ABB) rule, defined as follows

$$\alpha_k^{\mathrm{ABB}} = \begin{cases} \alpha_k^{\mathrm{BB2}} & \text{if } \frac{\alpha_k^{\mathrm{BB2}}}{\alpha_k^{\mathrm{BB1}}} < \tau, \ \tau \in (0,1) \\ \alpha_k^{\mathrm{BB1}} & \text{otherwise,} \end{cases}$$

which was modified by Frassoldati et al. in [9], introducing the so-called $\mathrm{ABB}_{\min}$ strategy:

$$\alpha_k^{\mathrm{ABB}_{\min}} = \begin{cases} \min\{\alpha_j^{\mathrm{BB2}} \mid j = \max(1, k - M_\alpha), \dots, k\} & \text{if } \frac{\alpha_k^{\mathrm{BB2}}}{\alpha_k^{\mathrm{BB1}}} < \tau, \ \tau \in (0,1) \\ \alpha_k^{\mathrm{BB1}} & \text{otherwise} \end{cases}$$

where $M_\alpha > 0$ is a prefixed integer.

The rationale behind the adaptive rule is to promote the use of smaller steplengths (realized by the BB2 rule), switching to a larger one when a reliable approximation of the inverse of the smallest eigenvalue is likely to occur. In this perspective, the $\mathrm{ABB}_{\min}$ strategy represents a refinement of this strategy, since it appears to provide a faster reduction of the gradient's component that would rather become dominant [15, 16].

Despite nonmonotonicity, BB related rules are shown to be much more efficient than (2) with surprisingly computational results even when generalized to nonquadratic and constrained problems [16–20]. We still do not have a complete understanding of their practical behavior; nevertheless some recent works investigating the relationships between the steplengths and the spectrum of the Hessian of the objective function were able to provide insight into the computational effectiveness of gradient methods, for both quadratic and general unconstrained/constrained optimization problems [12, 15, 21–25]. The design of effective steplength rules is the common ground on which all new gradient methods are based on.

In this paper, we propose a different approach to accelerate the convergence of gradient methods, based on the idea of performing, at selected iterations, minimization along descent directions other than the negative gradient $\mathbf{s}_k$, or even in affine subspaces of low dimension. Indeed, the idea of using composite search directions or minimization in low-dimensional subspaces has been explored by several authors in the past. The s-gradient method for convex quadratic functions was one of the first approaches to generalizing the steepest descent method using minimizations in spaces of dimension greater than one. Referring to the quadratic function

$$f(x) = \tfrac{1}{2}\mathbf{x}^\top A\mathbf{x},$$

in the $k-$th iteration the s-gradient method computes

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_1^k A\mathbf{x}_k + \alpha_2^k A^2\mathbf{x}_k + \cdots + \alpha_s^k A^s\mathbf{x}_k,$$

where the vector $\bar{\alpha}^k \equiv (\alpha_1^k, \alpha_2^k, \ldots, \alpha_s^k)^\top$ is the solution of the problem

$$\arg \min_{\alpha_1, \alpha_2, \ldots, \alpha_s} f(\mathbf{x}_k + \alpha_1 A\mathbf{x}_k + \alpha_2 A^2\mathbf{x}_k + \cdots + \alpha_s A^s\mathbf{x}_k).$$

The first work on this algorithm dates back to the early 1950s [26]. However, an elegant and comprehensive analysis of convergence and asymptotic behavior was presented by Forsythe [27]. In particular, generalizing the result demonstrated by Akaike [7] for $s = 1$, Forsythe shows that in the s-dimensional gradient method $\mathbf{x}_k$ eventually lies in a subspace of dimension at most $2s$. He also proves that, in general, the s-gradient process converges no faster than linearly. Recently, a more in-depth analysis was conducted in [28], based on a dynamical-system approach. Among other things, the authors demonstrate that an algorithm that periodically alternates between $s = 1$ and $s = 2$ significantly improves the convergence rate compared to $s = 2$, achieving superlinear convergence for $n = 3$. For general n, they conjecture that *"by switching periodically between algorithms one destroys the stability of the limiting behavior obtained when s is fixed (which is always associated with slow convergence)"*.

Different motivations underlie the "dog-leg" algorithm, proposed by Powell [29], which also incorporates a trust region condition. In the composite search direction of the form

$$\mathbf{s}_k = \alpha_k \mathbf{q}_k + \beta_k \mathbf{p}_k,$$

$\mathbf{p}_k$ represents a Newton, type Newton, or quasi-Newton direction, while $\mathbf{q}_k$ is the negative gradient. More generally, $\mathbf{q}_k$ can be chosen as a suitable descent direction designed to address non-convex problems or to achieve global convergence.

More recently, Bartholomew-Biggs has proposed a modified version of Newton's method based on composite directions, which involves fitting the objective function in the plane spanned by both the Newton direction and the steepest descent direction [30]. Additionally, a globalization strategy for Newton's method that combines Newton steps with gradient steps has been introduced in [31]. There are more examples of the use of composite search directions in the literature, including [32–34], and actually some classical methods such as the Polyak Heavy Ball Method [35], and the Conjugate Gradient method (see exercise 6.5 in [36]) can be framed as gradient-based minimization methods that use a composite search direction by combining the current gradient with the previous step. With the aim of designing an algorithm particularly suited to implementation on parallel computing systems, a generalization of the optimal gradient method to an s-dimensional conjugate gradient method is presented in [37]. Also the DWGM (Delayed Weighted Gradient Method) [38–40], although based on the idea of computing a new iteration as a weighted average between an intermediate new point and the penultimate iterate, can be seen as an algorithm using a composite search direction.

Many algorithms that utilize composite directions often use gradient methods to improve the convergence of second-order algorithms. However, our approach is different in that we aim to accelerate gradient methods by performing a minimization, at selected iterations, along descent directions explicitly chosen to act on the quadratic term of the objective function. We believe that this is an underexplored area that deserves more attention. To test our approach, we will focus primarily on the quadratic case (1), which provides an adequate basis for a preliminary analysis before addressing general minimization problems. For the latter, only a few very preliminary results are presented in this article.

The remainder of the paper is organized as follows. In Section 2 we introduce our acceleration strategy, preceded by some theoretical considerations that justify our choices. In particular, it is emphasized that attention should be paid to the reduction of the quadratic term and the linear term of the objective function in a descent method. In Section 3, we show how our strategy can be incorporated into a gradient projection algorithm for convex quadratic programming with bound constraints. In Section 4 we analyze through numerical experiments the performance of our acceleration strategy, both for unconstrained and bound constrained convex quadratic problems. In Section 5 we consider a possible generalization of our algorithm to the unconstrained minimization of a general function, with some preliminary computational results. Finally, in Section 6 we draw some conclusions.

Throughout the paper, vectors are denoted by boldface lowercase letters, $\|\cdot\|$ denotes the Euclidian norm, $\mathbf{g} = \mathbf{g}(x) = \nabla f(x)$. Given $\mathbf{v} \in \mathbb{R}^n$, $\mathcal{S}(\mathbf{v}) \subseteq \{1, ..., n\}$, the vector $\mathbf{v}^{\mathcal{S}}$ is defined componentwise as follows

$$\mathbf{v}^{\mathcal{S}}(i) := \begin{cases} \mathbf{v}(i) & \text{if } \; i \in \mathcal{S}(\mathbf{v}), \\ 0 & \text{if } \; i \notin \mathcal{S}(\mathbf{v}). \end{cases}$$

$\mathbf{0} \in \mathbb{R}^n$ indicates the null vector. Finally, we denote by $\kappa(A)$ the spectral condition number of the matrix A.

2 The Accelerated Gradient Method (AGM)

This section provides an analysis of the proposed method along with some theoretical results. To describe our approach, we start with some considerations about the linear term and the quadratic term of the objective function in (1).

If we define the error vector at iteration k as

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^*,$$

in a gradient method, where $\mathbf{x}^* = A^{-1}\mathbf{b}$ is the solution of problem (1), the evolution of the error follows the rule

$$\mathbf{e}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}^* = (I - \alpha_k \; A) \; \mathbf{e}_k,$$

which is governed entirely by the matrix $I - \alpha_k A$. The linear term $-\mathbf{b}^\top \mathbf{x}$ does not affect the contraction factors in the error dynamics. Instead, the vector $\mathbf{b}$ only influences $\mathbf{x}^*$

by shifting its position away from the origin. The convergence rate of gradient descent is dictated by the spectrum of A. Since A is symmetric positive definite, it can be diagonalized as

$$A = V \Lambda V^\top,$$

where V is an orthogonal matrix and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ contains the eigenvalues of A. Let us define the transformed error

$$\mathbf{y}_k = V^\top \mathbf{e}_k.$$

Then, the error recursion in the eigenbasis becomes

$$\mathbf{y}_{k+1} = (I - \alpha\ \Lambda)\ \mathbf{y}_k,$$

or, componentwise,

$$\mathbf{y}_{k+1}(i) = (1 - \alpha\ \lambda_i)\ \mathbf{y}_k(i), \quad i = 1, \ldots, n,$$

the decay along each eigen-direction is given by $|1 - \alpha\lambda_i|$, completely indipendent of $\mathbf{b}$.

We notice that the solution $\mathbf{x}^*$ of problem (1) satisfies the equation

$$\mathbf{b}^\top \mathbf{x} = \mathbf{x}^\top A \mathbf{x}, \tag{3}$$

then, if $\mathbf{x}_k \to \mathbf{x}^*$, then

$$\lim_{k \to +\infty} \mathbf{b}^\top \mathbf{x}_k = \lim_{k \to +\infty} \mathbf{x}_k{}^\top A \mathbf{x}_k = \phi^*$$

for some $\phi^* > 0$.

The following proposition characterizes the solution to problem (1) in the set of points that satisfy the equation (3).

Proposition 1 *Solving the problem* (1) *is equivalent to solving the following constrained maximization problem*

$$\begin{aligned} \max \tilde{f}(\mathbf{x}) &\equiv \mathbf{x}^\top A \mathbf{x} \\ \text{s.t. } h(\mathbf{x}) &\equiv \mathbf{x}^\top A \mathbf{x} - \mathbf{b}^\top \mathbf{x} = 0. \end{aligned} \tag{4}$$

Proof The strict convexity of problem (4) ensures uniqueness of the optimal solution. The thesis follows from the observation that

the Lagrange conditions for (4)

$$2A\mathbf{x} = \lambda\ (2A\mathbf{x} - \mathbf{b}),$$

$$\mathbf{x}^\top A \mathbf{x} - \mathbf{b}^\top \mathbf{x} = 0$$

are satisfied by $\lambda = 2$ and $\mathbf{x}^* = A^{-1}\mathbf{b}$.

□

Therefore, the linear and quadratic terms are perfectly balanced in the solution, and the latter reaches the maximum value in the feasibility set of problem (4). The functions $f(\mathbf{x})$, $\tilde{f}(\mathbf{x}), h(\mathbf{x})$, are depicted in Figure 1.

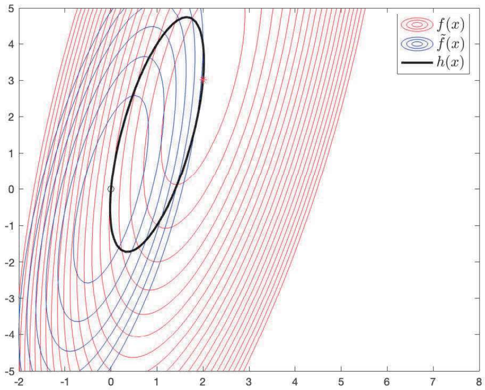

Fig. 1 Graphical representation of the problem (4).

Proposition 2 *Let us assume that* $\mathbf{x}_k$ *is such that*

$$\mathbf{x}_k^\top A\mathbf{x}_k < \mathbf{b}^\top \mathbf{x}_k \ \ [\mathbf{x}_k^\top A\mathbf{x}_k > \mathbf{b}^\top \mathbf{x}_k], \tag{5}$$

then $\mathbf{d}_k = \mathbf{x}_k$ $[\mathbf{d}_k = -\mathbf{x}_k]$ *is*

(a) a descent direction for $f(\mathbf{x})$ *at* $\mathbf{x}_k$*,*
(b) an ascent [descent] direction for $\tilde{f}(\mathbf{x})$ *at* $\mathbf{x}_k$*.*

Proof Because of (5) one has $\nabla f(\mathbf{x_k})^\top \mathbf{x_k} < \mathbf{0}$ $[\nabla \mathbf{f}(\mathbf{x_k})^\top(-\mathbf{x_k}) < \mathbf{0}]$, which proves (a). While (b) follows from the positive definitiveness of A. □

Proposition 2 shows that if $\mathbf{x}_{k+1}$ is computed by minimizing $f(\mathbf{x})$ along $\mathbf{x}_k$, moving from $\mathbf{x}_k$, this will not only produce a reduction of $f(\mathbf{x})$ but also create a "balancing effect" on the quadratic part of the objective function. Specifically, the quadratic term will decrease if $\mathbf{x}_k^\top A\mathbf{x}_k > \mathbf{b}^\top \mathbf{x}_k$, and it will increase otherwise.

Here, we present some simple theoretical results that can help us better understand the behavior of the two sequences $\{\mathbf{b}^\top \mathbf{x}_k\}$ and $\{\mathbf{x}_k^\top A\mathbf{x}_k\}$, hereby providing a convincing motivation for our acceleration strategy.

Proposition 3 *Let assume* $\mathbf{b} \neq 0$*, then for all* $\mathbf{x} \in \mathbb{R}^n$ *such that* $\mathbf{b}^\top \mathbf{x} \neq 0$ *one has*

$$\mathbf{x}^\top A\mathbf{x} \leq \kappa(A) \frac{\|\mathbf{x}\|}{\|\mathbf{x}^*\|} \frac{|\mathbf{b}^\top \mathbf{x}|}{|\cos\theta|}$$

where $\mathbf{x}^*$ *is the solution of* (1) *and* θ *is the angle between* $\mathbf{b}$ *and* $\mathbf{x}$*.*

Proof Notice that from $\mathbf{b} \neq 0$ it follows $\|\mathbf{x}^*\| \neq 0$.

$$\mathbf{x}^\top A\mathbf{x} \;\leq \|A\|\|\mathbf{x}\|^2 = \|A\|\frac{\|\mathbf{x}\|^2}{\|\mathbf{x}^*\|}\|\mathbf{x}^*\|$$
$$= \|A\|\frac{\|\mathbf{x}\|^2}{\|\mathbf{x}^*\|}\|A^{-1}\mathbf{b}\| \leq \|A\|\frac{\|\mathbf{x}\|}{\|\mathbf{x}^*\|}\|A^{-1}\|\frac{|\mathbf{b}^\top\mathbf{x}|}{|\cos\theta|} = \kappa(A)\frac{\|\mathbf{x}\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x}|}{|\cos\theta|}.$$

□

Proposition 4 *Let us assume that the hypotheses of Proposition 3 hold, then for all $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{b}^\top\mathbf{x}^* \neq \mathbf{b}^\top\mathbf{x}$, we have*

$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*\top} A\mathbf{x}^*| \leq \kappa(A)\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{|\cos\theta|}, \quad (4A)$$
$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*\top} A\mathbf{x}^*| \leq \|A\|\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{b}\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{|\cos\theta|}, \quad (4B)$$

where θ is the angle between $\mathbf{b}$ and $\mathbf{x} - \mathbf{x}^*$.

Proof

(4A)
$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*\top} A\mathbf{x}^*| = |(\mathbf{x}-\mathbf{x}^*)^\top A(\mathbf{x}+\mathbf{x}^*)| \leq \|A\|\|\mathbf{x}+\mathbf{x}^*\|\|\mathbf{x}-\mathbf{x}^*\| \leq$$
$$\leq \|A\|\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{\|\mathbf{b}\||\cos\theta|}\|\mathbf{x}^*\| = \|A\|\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{\|\mathbf{b}\||\cos\theta|}\|A^{-1}\mathbf{b}\|$$
$$\leq \kappa(A)\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{|\cos\theta|},$$

(4B)
$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*\top} A\mathbf{x}^*| = |(\mathbf{x}-\mathbf{x}^*)^\top A(\mathbf{x}+\mathbf{x}^*)| \leq \|A\|\|\mathbf{x}+\mathbf{x}^*\|\|\mathbf{x}-\mathbf{x}^*\| =$$
$$= \|A\|\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{b}\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{|\cos\theta|}.$$

□

About the inequalities (4A - 4B) in Proposition 4 let us consider the case of $A \in \mathbb{R}^{n\times n}$ symmetric such that $\lambda = \lambda_{\max}(A) \geq \lambda_{\min}(A) = 1$, $\mathbf{b} = \lambda\mathbf{v}$ where $\mathbf{v}$ is such that $A\mathbf{v} = \lambda\mathbf{v}$, so that $\mathbf{x}^* = \mathbf{v}$; for any $\mathbf{x} = \beta\mathbf{v}$ with $\beta > 0,\ \beta \neq 1$

$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*\top} A\mathbf{x}^*| = |(\mathbf{x}-\mathbf{x}^*)^\top A(\mathbf{x}+\mathbf{x}^*)| = |(\mathbf{x}-\mathbf{x}^*)^\top(\lambda)(\mathbf{x}+\mathbf{x}^*)| =$$
$$\lambda\|\mathbf{x}-\mathbf{x}^*\|\|\mathbf{x}+\mathbf{x}^*\|,$$

$$\kappa(A)\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{x}^*\|}\frac{|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*|}{|\cos\theta|} = \|A\|\frac{\|\mathbf{x}+\mathbf{x}^*\|}{\|\mathbf{b}\||\cos\theta|}|\mathbf{b}^\top\mathbf{x} - \mathbf{b}^\top\mathbf{x}^*| = \lambda\|\mathbf{x}-\mathbf{x}^*\|\|\mathbf{x}+\mathbf{x}^*\|,$$

i.e. (4A - 4B) are satisfied as equality. Note that in this case it holds

$$|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*T} A\mathbf{x}^*| = \kappa(A)\|\mathbf{x}-\mathbf{x}^*\|\|\mathbf{x}+\mathbf{x}^*\|.$$

In general, the quadratic term is governed by the eigenstructure of A, and for ill-conditioned problems $|\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^{*T} A\mathbf{x}^*|$ can be significantly large even when $\|\mathbf{x} - \mathbf{x}^*\|$ is small.

The strategy we propose aims to balance the linear and quadratic terms. We suggest intervening on the latter by computing, at selected iterations, acceleration steps into a standard gradient scheme (1).

Given $\mathbf{x}_k$, an acceleration step computes next iteration by one of the following rules:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k - \beta_k \mathbf{x}_k, \text{ where } (\alpha_k, \beta_k) = \mathbf{argmin}_{\alpha,\beta} f(\mathbf{x}_k - \alpha \mathbf{g}_k - \beta \mathbf{x}_k) \quad [\mathbf{2D}]$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta_k \mathbf{x}_k, \text{ where } \beta_k = \mathbf{argmin}_{\beta} f(\mathbf{x}_k - \beta \mathbf{x}_k). \quad [\mathbf{1D}]$$

We notice that the Newton step for the quadratic term $\mathbf{x}_k A \mathbf{x}_k$ is $-\mathbf{x}_k$, which is also a descent direction for $f(\mathbf{x})$ if

$$\mathbf{x}_k^\top \mathbf{g}_k > 0. \tag{6}$$

Notice that, if (6) holds, in accordance with Proposition 2, a [1D] step will produce a decrease for both $f(\mathbf{x})$ and $\tilde{f}(\mathbf{x})$. A [2D] step performs the minimization in the affine subspace generated by the gradient at $\mathbf{x}_k$ and the Newton direction (or its opposite) of the pure quadratic function $\mathbf{x}^\top A\mathbf{x}$. It is worthwhile to note that the [2D] step can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \left(\left(A + \frac{\beta_k}{\alpha_k} I \right) \mathbf{x}_k - \mathbf{b} \right), \tag{7}$$

i.e. a gradient step of the function $f(\mathbf{x})$ in which the spectrum of the Hessian has been shifted. Therefore, to a certain extent, [2D] can be interpreted as a gradient for the regularised function $f(x) + \frac{\beta_k}{\alpha_k}\|\mathbf{x}\|$ in the Tikhonov sense, provided that $\frac{\beta_k}{\alpha_k} > 0$.

In our proposed scheme, an acceleration step is performed whenever

$$\frac{|\mathbf{x}_k^\top \mathbf{g}_k|}{\|\mathbf{x}_k\| \|\mathbf{g}_k\|} \geq \gamma, \tag{8}$$

for some $\gamma \in (0, 1)$, or $\texttt{mod}(k, \mu) = 0$. Here, k is the iteration number, μ is some prefixed positive integer and `mod` is the modulus after division. In this way, the algorithm is forced to perform at least an acceleration step every μ iterations (with $\mu = \infty$ meaning that no such forcing is applied). The condition (8) ensures that $\mathbf{x}_k$ is a search direction "sufficiently" not orthogonal to the gradient. We also note that the algorithm starts with an acceleration step, provided that $\mathbf{x}_0^\top \mathbf{g}_0 > 0$. The purpose of this initial step is to promote a balance between linear and quadratic terms from the beginning.

About the solution of the minimization problem [**2D**], we note that α_k and β_k are solutions of the following 2×2 system, where, for the sake of simplicity, we omit the iterations indices

$$\begin{bmatrix} \mathbf{g}^\top A\mathbf{g} & \mathbf{g}^\top A\mathbf{x} \\ \mathbf{g}^\top A\mathbf{x} & \mathbf{x}^\top A\mathbf{x} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{g}^\top \mathbf{g} \\ \mathbf{x}^\top \mathbf{g} \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & \frac{\mathbf{g}^\top A\mathbf{x}}{\mathbf{g}^\top A\mathbf{g}} \\ \frac{\mathbf{g}^\top A\mathbf{x}}{\mathbf{x}^\top A\mathbf{x}} & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top A\mathbf{g}} \\ \frac{\mathbf{x}^\top \mathbf{g}}{\mathbf{x}^\top A\mathbf{x}} \end{bmatrix}$$

i.e.

Algorithm 1 Accelerated Gradient Method (**AGM**)

```
1:  Initialization: $\mathbf{x}_0 \in \mathbb{R}^n$; $k = 0$; $\gamma \in (0,1)$; $\text{tol} > 0$; $\text{maxiter}, \mu \in \mathbb{N}$;
2:  $\mathbf{g}_0 = A\mathbf{x}_0 - \mathbf{b}$;
3:  while $\|\mathbf{g}_k\| \geq \text{tol} \cdot \|\mathbf{g}_0\|$ and $k \leq \text{maxiter}$ do
4:      $k = k + 1$;
5:      if $\frac{|\mathbf{x}_k^\top \mathbf{g}_k|}{\|\mathbf{x}_k\|\|\mathbf{g}_k\|} > \gamma$ or $\mathtt{mod}(k,\mu) = 0$ or $(k = 1$ and $\mathbf{x}_k^\top \mathbf{g}_k > 0)$ then
6:          Either:
7:          $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k - \beta_k \mathbf{x}_k$
8:          where $(\alpha_k, \beta_k) = \mathbf{argmin}_{\alpha,\beta} f(\mathbf{x}_k - \alpha \mathbf{g}_k - \beta \mathbf{x}_k)$    [2D]
9:          $\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k A\mathbf{g}_k - \beta_k A\mathbf{x}_k$
10:         Or:
11:         $\mathbf{x}_{k+1} = \mathbf{x}_k - \beta_k \mathbf{x}_k$
12:         where $\beta_k = \text{argmin}_\beta f(\mathbf{x}_k - \beta \mathbf{x}_k) = 1 - \frac{\mathbf{b}^\top \mathbf{x}_k}{\mathbf{x}_k A\mathbf{x}_k}$    [1D]
13:         $\mathbf{g}_{k+1} = \mathbf{g}_k - \beta_k A\mathbf{x}_k$
14:     else
15:         $\alpha_k > 0$ is computed by any convergent GM stepsize rule
16:         $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
17:         $\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k A\mathbf{g}_k$
18:     end if
19: end while
```

$$\alpha = \frac{(\mathbf{x}^\top A\mathbf{x})(\mathbf{g}^\top \mathbf{g}) - (\mathbf{g}^\top A\mathbf{x})(\mathbf{x}^\top \mathbf{g})}{(\mathbf{g}^\top A\mathbf{g})(\mathbf{x}^\top A\mathbf{x}) - (\mathbf{g}^\top A\mathbf{x})^2},$$

$$\beta = \frac{(\mathbf{g}^\top A\mathbf{g})(\mathbf{x}^\top \mathbf{g}) - (\mathbf{g}^\top A\mathbf{x})(\mathbf{g}^\top \mathbf{g})}{(\mathbf{g}^\top A\mathbf{g})(\mathbf{x}^\top A\mathbf{x}) - (\mathbf{g}^\top A\mathbf{x})^2}.$$

In the AGM algorithm, each step can be computed using only one matrix vector product. Specifically, if strategy [2D] is performed, $A\mathbf{x}_k$ can be computed in the gradient update as $\mathbf{g}_k + \mathbf{b}$, therefore $A\mathbf{g}_k$ is the only matrix-vector product that needs to be computed.

We observe that AGM inherits the convergence properties of GM, provided that the number of acceleration steps is finite or, if infinite, provided that an iteration of a gradient scheme satisfying the assumption in [36, Prop. 1.2.6], occurs infinitely often. Indeed, in this case, the convergence of AGM to the solution $\mathbf{x}^*$ of problem (1) can be derived by the following proposition.

Proposition 5 *Let $f(\mathbf{x})$ be a strictly convex quadratic function. Let $\{\mathbf{x}_k\}$ be an infinite sequence generated by Algorithm 1, where the steplength α_k at the step 15 of the Algorithm 1 is chosen by the exact minimization rule (2) or the Armijo rule [36, eq. (1.11)], then the following properties hold:*

(a) the sequence $\{\mathbf{x}_k\}$ converges to the unique solution $\mathbf{x}^*$ of problem 1;
(b) the sequence $\{f(\mathbf{x}_k)\}$ has Q-linear convergence rate.

Proof (*a*). Let first observe that $\{f(\mathbf{x}_k)\}$ is a nonincreasing sequence, i.e.

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k), \quad k = 0, 1, \dots,$$

and the sequence $\{\mathbf{x}_k\} \subseteq \{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ is a bounded sequence, due to the strict convexity of the objective function f. If $\mathcal{K}_{\mathrm{GM}}$ is the set of indices corresponding to the iterates generated by the gradient method, $\mathcal{K}_{\mathrm{GM}}$ is infinite, the subsequence $\{\mathbf{x}_k : k \in \mathcal{K}_{\mathrm{GM}}\}$ is bounded and it admits at least one limit point. Then, from [36, Prop. 1.2.6] it follows that every limit point of the subsequence $\{\mathbf{x}_k : k \in \mathcal{K}_{\mathrm{GM}}\}$ is a stationary point. Since f is strictly convex, the solution $\mathbf{x}^*$ is the only stationary point of problem (1). As a consequence, there is a subsequence of $\{\mathbf{x}_k\}$ that converges to $\mathbf{x}^*$ and $\{f(\mathbf{x}_k)\}$ converges to $\{f(\mathbf{x}^*)\}$. In particular, we may write

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \nabla f(\mathbf{x}^*)^\top (\mathbf{x}_k - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^\top \nabla^2 f(\mathbf{x}^*)(\mathbf{x}_k - \mathbf{x}^*) \geq \frac{\lambda_{\min}}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2.$$

Since $\{f(\mathbf{x}_k)\}$ converges to $\{f(\mathbf{x}^*)\}$, the previous inequality implies that $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^*$.
(b). Let $\mathbf{x}_{k+1}$ be the point generated from $\mathbf{x}_k$ via the acceleration step [**2D**] (or equiv. [**1D**]), whereas let denote by $\bar{\mathbf{x}}_{k+1}$ the point generated from $\mathbf{x}_k$ via the exact minimization rule. By definition, we have

$$f(\mathbf{x}_{k+1}) \leq f(\bar{\mathbf{x}}_{k+1})$$

where the strict inequality holds for $\beta > 0$. Then, from [36, Prop. 1.3.1], we have

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}_{k+1}) - f(\mathbf{x}^*) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^2 (f(\mathbf{x}_k) - f(\mathbf{x}^*)).$$

Let now consider the case where the stepsize α_k at Step 15 of Algorithm 1 is determined by the Armijo rule. If $\bar{\mathbf{x}}_{k+1}$ is the iterate generated via the Armijo rule, by using a similar argument to the previous case, one can conclude

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}_{k+1}) - f(\mathbf{x}^*) \leq \left(1 - \frac{4\lambda_{\min}\beta\sigma(1-\sigma)}{\lambda_{\max}}\right)(f(\mathbf{x}_k) - f(\mathbf{x}^*)).$$

where the second inequality follows from [36, 1.3.11]. □

In cases where the earlier theoretical result is not applicable, performing a finite number of acceleration steps ensures that the convergence behavior of the underlying gradient methods is recovered.

To gain a preliminary understanding of the impact of acceleration steps in gradient methods, such as the Cauchy SD method and the Relaxed Cauchy SD (RSD) method [41], we examined two different randomly generated problems with $n = 500$ variables, and $\kappa(A)$ equal to $10^3, 10^6$, respectively. For these problems, we set $\gamma = 0.5$, $\mu = \infty$, $\text{maxiter} = 5000\,, \text{tol} = 10^{-5}$. For each problem, we considered two different starting points with random components in $[0, 1]$: $\mathbf{x}_0^+$, such that $(\mathbf{x}_0^+)^\top A\mathbf{x}_0^+ > \mathbf{b}^\top \mathbf{x}_0^+$, and $\mathbf{x}_0^-$ such that $(\mathbf{x}_0^-)^\top A\mathbf{x}_0^- < \mathbf{b}^\top \mathbf{x}_0^-$. In the following, the prefix “A” will denote the accelerated algorithms; subscript 1 or 2 will be used according to whether step [1D] or [2D] is used. A primary objective of this numerical experiment is to verify the ability of steps [1D] and [2D] to promote the balance between the linear and quadratic terms of the objective function. It is therefore interesting to compare the practical behavior of the two sequences $\{\mathbf{b}^\top \mathbf{x}_k\}$ and $\{\mathbf{x}_k^\top A\mathbf{x}_k\}$. In Figure 2 and Figure 3 is reported the convergence history for the first iterations of the two sequences for the different algorithms.

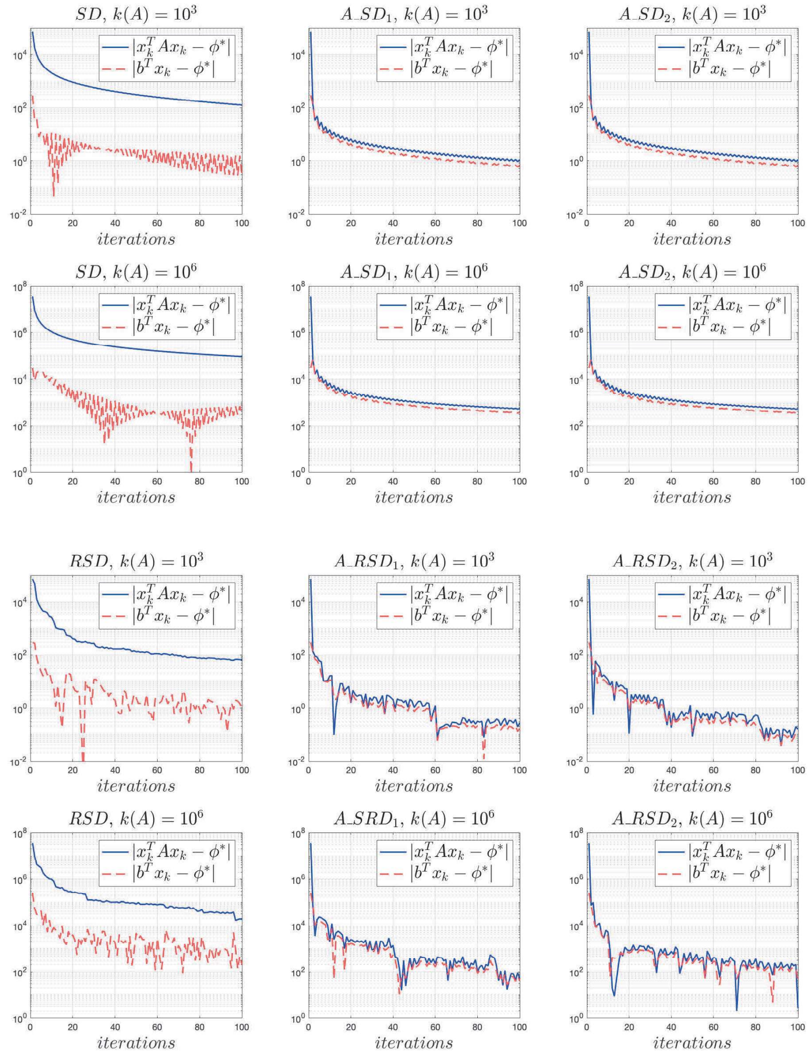

Fig. 2 Convergence history for Linear and quadratic terms: Accelerated vs Standard algorithms, starting point $\mathbf{x}_0^+$.

The impact of utilizing the [2D] and [1D] steps is quite significant, effectively driving the reduction of the gap between the quadratic term and the linear term in very few iterations. This is particularly evident in Figure 2, which refers to the case where $\mathbf{x}_0^+$ is chosen as the starting point. In our view, this is the most significant case, as at a randomly chosen point one would expect the quadratic term to dominate the linear term. In this case, the gap between the two terms is quite evident, with the quadratic

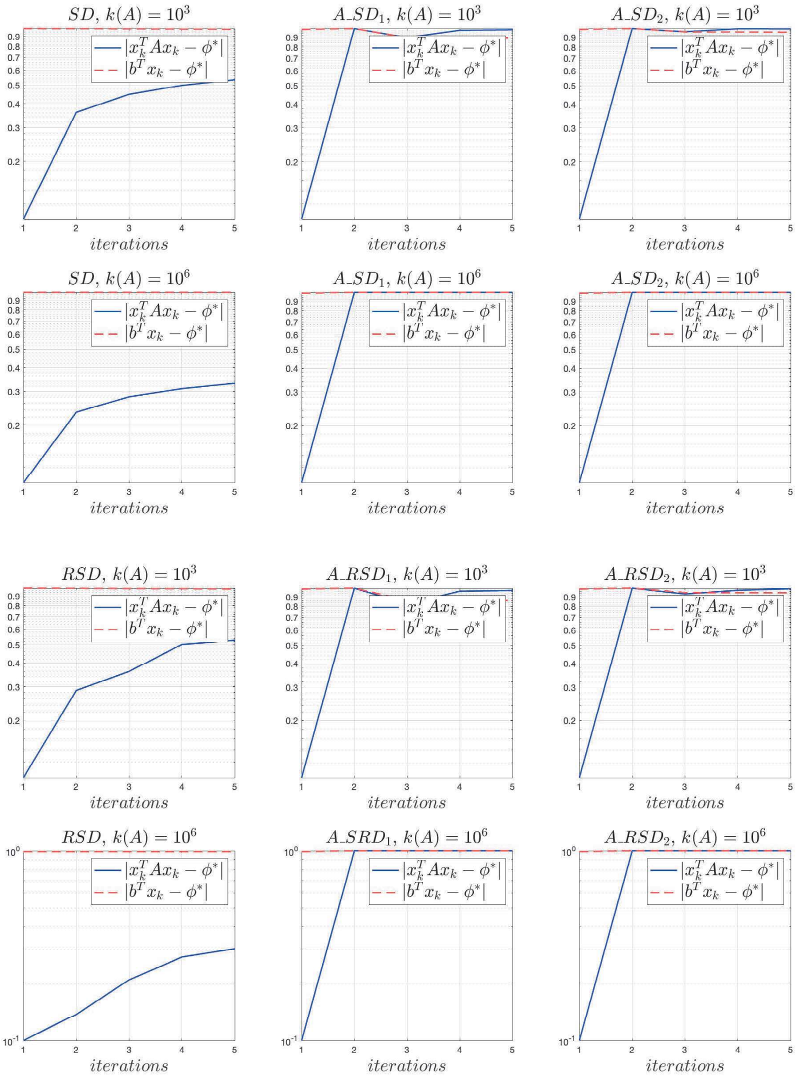

Fig. 3 Convergence history for Linear and quadratic terms: Accelerated vs Standard algorithms, starting point $\mathbf{x}_0^-$.

term much larger, and both Algorithm SD and Algorithm RSD tend to reduce this gap very slowly. As expected, ill-conditioning accentuates this phenomenon.

Regarding the convergence of the algorithms under consideration, with starting point $\mathbf{x}_0^+$ to the solution, an overall picture is provided in Figure 4. It should be noted that $\mathbf{0}$ was also tested as a starting point for the standard algorithms (denoted by the suffix (0)). This is because this choice ensures a perfect balance between the quadratic and linear terms. In practice, however, it proves to be a poor choice. For the problems under consideration, the use of acceleration steps significantly improves the convergence compared to the underlying gradient method. However, a complete picture of the performance of the different algorithms, also depending on the choice of the starting point, will be given by the extensive numerical experiments in Section 4.

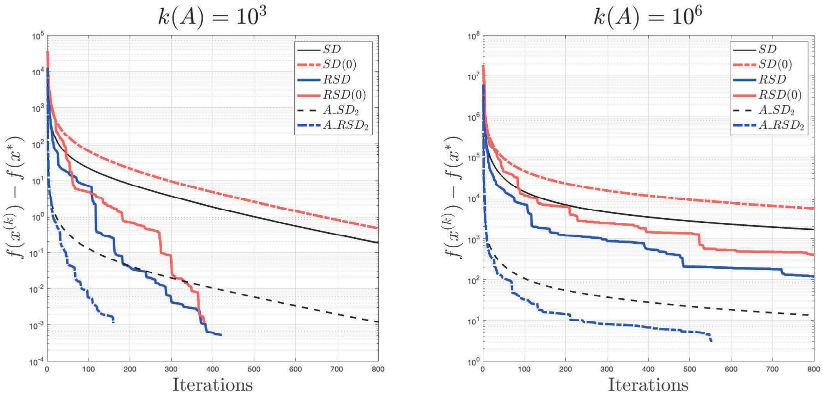

Fig. 4 Convergence history (first 800 iterations): standard gradient methods vs accelerated gradient methods.

3 The bound constrained case

We will now examine the application of acceleration steps in constrained optimization problems. Specifically, we will focus on the bound-constrained convex quadratic problem (BCQP)

$$\begin{aligned} \min\ & f(\mathbf{x}) \equiv \tfrac{1}{2}\, \mathbf{x}^\top A\, \mathbf{x} - \mathbf{b}^\top \mathbf{x}, \\ \text{s.t. } & \mathbf{x} \in \Omega = \{\mathbf{l} \le \mathbf{x} \le \mathbf{u}\}, \end{aligned} \tag{9}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{l} \in \{\mathbb{R} \cup \{-\infty\}\}^n$, $\mathbf{u} \in \{\mathbb{R} \cup \{+\infty\}\}^n$, and, without loss of generality, $\mathbf{l}(i) < \mathbf{u}(i)$ for all i. For any $\mathbf{x} \in \Omega$ we define the following index sets

$$\mathcal{A}_l(\mathbf{x}) := \{i : \mathbf{x}(i) = \mathbf{l}(i)\},\ \mathcal{A}_u(\mathbf{x}) := \{i : \mathbf{x}(i) = \mathbf{u}(i)\},\ \mathcal{A}(\mathbf{x}) := \mathcal{A}_l(\mathbf{x}) \cup \mathcal{A}_u(\mathbf{x}),$$
$$\mathcal{B}(\mathbf{x}) := \{i : (i \in \mathcal{A}_l(\mathbf{x}) \wedge \nabla f_i(\mathbf{x}) \ge 0)\ \vee\ (i \in \mathcal{A}_u(\mathbf{x})\ \wedge\ \nabla f_i(\mathbf{x}) \le 0)\},$$
$$\mathcal{F}(\mathbf{x}) := \{1, \ldots, n\} \setminus \mathcal{A}(\mathbf{x}),\ \mathcal{W}(\mathbf{x}) := \{1, \ldots, n\} \setminus \mathcal{B}.$$

$\mathcal{A}(\mathbf{x})$, $\mathcal{B}(\mathbf{x})$, $\mathcal{F}(\mathbf{x})$ and $\mathcal{W}(\mathbf{x})$ are called the active, the binding, the free and the working sets at $\mathbf{x}$, respectively.

The concept of binding, or strongly active variable, is of fundamental importance for optimality conditions and has been generalized to the case of linearly constrained quadratic problems in [42]. The optimality conditions for linearly constrained problems can be stated componentwise in terms of binding components: $\mathbf{x}^*$ is solution of (9) if and only if $\nabla f_i(\mathbf{x}^*) = 0$ for any $i \in \mathcal{F}(\mathbf{x}^*)$ and $\mathcal{B}(\mathbf{x}^*) = \mathcal{A}(\mathbf{x}^*)$. In order to simplify the notation, we also define

$$\mathcal{A}^* = \mathcal{A}(\mathbf{x}^*), \quad \mathcal{A}_l^* = \mathcal{A}_l(\mathbf{x}^*), \quad \mathcal{A}_u^* = \mathcal{A}_u(\mathbf{x}^*).$$

Given $\rho, \sigma \in (0,1)$, a generic iteration of the Gradient Projection (GP) method for problem (9), equipped with an Armijo linesearch rule, can be defined alternatively as

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_k^P[\theta, \nu, \mathbf{d}] = \mathbf{P}_\Omega(\mathbf{x}_k + \theta\nu\mathbf{d}), \tag{10}$$

or

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_k^L[\theta, \nu, \mathbf{d}] = \mathbf{x}_k + \theta(\mathbf{P}_\Omega(\mathbf{x}_k + \nu\mathbf{d}_k) - \mathbf{x}_k), \tag{11}$$

where $\nu > 0$ is a steplength and $\theta \in (0,1)$ is the linesearch parameter. In particular, θ is selected via a backtracking strategy as $\theta = \rho^h$, where h is the first nonnegative integer such that

$$f(\hat{\mathbf{x}}_k[\rho^h, \nu, \mathbf{d}_k]) \leq f(\mathbf{x}_k) + \sigma \nabla f(\mathbf{x}_k)^\top (\hat{\mathbf{x}}_k[\rho^h, \nu, \mathbf{d}_k] - \mathbf{x}_k).$$

The projected-search in (10) explores a path that is piecewise-linear and continuous. This path is generated by projecting the search direction $\mathbf{d}_k$ onto the feasible region Ω. In contrast, the line search method in (11) searches along the direction given by $\mathbf{d}_k = \mathbf{P}_\Omega(\mathbf{x}_k + \nu\mathbf{d}_k) - \mathbf{x}_k$. For the Gradient Projection (GP) method, convergence can be assured regardless of which of these two strategies is implemented [43]. However, if a descent direction $\mathbf{d}_k$ is selected that differs from the antigradient, the situation becomes considerably more complex. In such cases, achieving even a simple decrement along the search path is not straightforward. The two examples illustrated in Figure 5 provide a clearer insight into this issue. In the case shown on the left, the linear path does not allow for any decrease in the objective function, while the piecewise path does. In the case depicted on the right, neither path permits any displacement from $\mathbf{x}$.

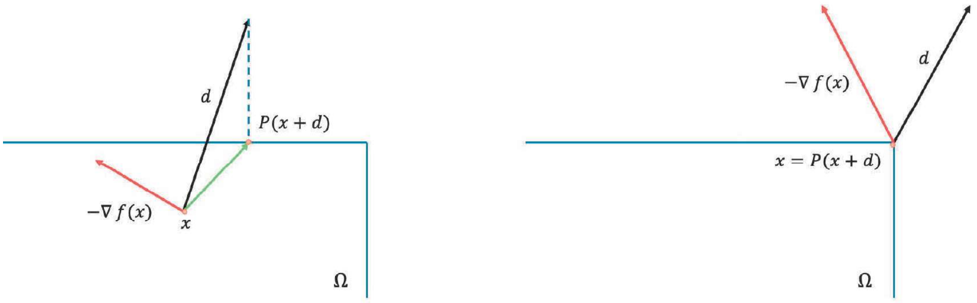

Fig. 5 Sketch of projected paths.

Algorithm 2, called APGM, is a variant of the AGM algorithm adapted to convex quadratic problems with bound constraints. Specifically, APGM is a very general GP method, where acceleration steps are computed at selected iterations and involve only those variables whose indices belong to the current working set.

Algorithm 2 Accelerated Projected Gradient Method (APGM)

```
**Initialization:** $\mathbf{x}_0 \in \mathbb{R}^n$; $k = 0$; $\gamma, \sigma \in (0,1)$; $0 < \alpha_{\min} < \alpha_{\max}$; $M \geq 1$;
    maxiter, $\mu \in \mathbb{N}$; tol $> 0$;
$\mathbf{g}_0 = A\mathbf{x}_0 - \mathbf{b}$;
**while** $\|\mathbf{g}_k\| \geq \text{tol} \cdot \|\mathbf{g}_0\|$ **and** $k \leq \text{maxiter}$ **do**
    $k = k + 1$;
    **if** $\frac{(\mathbf{x}_k^{\mathcal{W}})^\top \mathbf{g}_k^{\mathcal{W}}}{\|\mathbf{x}_k^{\mathcal{W}}\| \|\mathbf{g}_k^{\mathcal{W}}\|} > \gamma$ **or** $\texttt{mod}(\mathrm{k}, \mu) = 0$ **or** ($k = 1$ **and** $(\mathbf{x}_k^{\mathcal{W}})^\top \mathbf{g}_k^{\mathcal{W}} > 0$) **then**
        **Either:**
        $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k - \beta_k \mathbf{x}_k$
        where $(\alpha_k, \beta_k) = \mathbf{argmin}_{\alpha,\beta} f(\mathbf{x}_k - \alpha \mathbf{g}_k^{\mathcal{W}} - \beta \mathbf{x}_k^{\mathcal{W}})$ **[2D]**
        $\mathbf{d}_k = \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k$
        $\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k + \rho^h \mathbf{d}_k)$
        where $h$ is the first nonnegative integer such that
        $f(P_\Omega(\mathbf{x}_k + \rho^h \mathbf{d}_k)) < f(\mathbf{x}_k)$
        **Or:**
        $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k - \beta_k \mathbf{x}_k$
        where $(\alpha_k, \beta_k) = (0, \mathbf{argmin}_{\alpha,\beta} f(\mathbf{x}_k - \beta \mathbf{x}_k^{\mathcal{W}})) = 1 - \frac{\mathbf{b}^\top \mathbf{x}_k^{\mathcal{W}}}{(\mathbf{x}_k^{\mathcal{W}})^\top A \mathbf{x}_k^{\mathcal{W}}}$ **[1D]**
        $\mathbf{d}_k = \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k$
        $\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k + \rho^h \mathbf{d}_k)$
        where $h$ is the first nonnegative integer such that
        $f(P_\Omega(\mathbf{x}_k + \rho^h \mathbf{d}_k)) < f(\mathbf{x}_k)$
    **else**
        $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ is computed by any stepsize rule
        $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$
        $\mathbf{d}_k = P_\Omega(\tilde{\mathbf{x}}_{k+1}) - \mathbf{x}_k$
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \rho_k \mathbf{d}_k$
        where the steplength $\rho_k$ satisfies the generalized Armijo condition [44]:
        $f(\mathbf{x}_k + \rho_k \mathbf{d}_k) \leq \max_{0 \leq j \leq \{k, M-1\}} f(\mathbf{x}_{k-j}) + \sigma \rho_k \mathbf{d}_k^\top \mathbf{g}_k$
    **end if**
**end while**
```

We note that, in the acceleration steps, $\mathbf{x}_{k+1}$ is computed via a line search along the piecewise-linear path $\hat{\mathbf{x}}_k^P[\rho]$, rather than the linear path $\hat{\mathbf{x}}_k^L[\rho]$. This is because the latter is not guaranteed to satisfy a descent condition, whereas for sufficiently small $\rho > 0$ one has $f(\mathbf{x}_k^P[\rho]) < f(\mathbf{x}_k)$, as established in the following proposition.

Proposition 6 *Let $\mathbf{x} \in \Omega$, $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$ with $\mathbf{d}(i) = 0$ if $i \in \mathcal{B}(\mathbf{x})$. If $f(\mathbf{x} + \mathbf{d}) \leq f(\mathbf{x})$, then there exists $\bar{\rho} > 0$ such that $f(\mathbf{x}[\rho]) < f(\mathbf{x})$ for $\rho \in (0, \bar{\rho})$, where $\mathbf{x}[\rho] := P_\Omega(\mathbf{x} + \rho \mathbf{d})$.*

Proof If we define the index set $\mathcal{I}(\mathbf{x})$ as follows:

$$i \in \mathcal{I}(\mathbf{x}) \text{ if one of the} \atop \text{following conditions holds} \begin{cases} \mathbf{x}(i) \in (\mathbf{l}(i), \mathbf{u}(i)) & \text{and } i \in \mathcal{W}(\mathbf{x}), \\ \mathbf{x}(i) = \mathbf{l}(i),\ \mathbf{d}(i) > 0 & \text{and } i \in \mathcal{W}(\mathbf{x}), \\ \mathbf{x}(i) = \mathbf{u}(i),\ \mathbf{d}(i) < 0 & \text{and } i \in \mathcal{W}(\mathbf{x}), \end{cases}$$

then, for $\rho > 0$ sufficiently small

$$\mathbf{x}[\rho](i) = \begin{cases} \mathbf{x}(i) + \rho \mathbf{d}(i) & \text{if } i \in \mathcal{I}(\mathbf{x}), \\ \quad \mathbf{x}(i) & \text{if } i \notin \mathcal{I}(\mathbf{x}). \end{cases}$$

Note that, because of the convexity of the objective function one has

$$0 > \nabla f(\mathbf{x})^\top \mathbf{d} = \sum_{i \in \mathcal{I}(\mathbf{x})} \mathbf{g}(i)\mathbf{d}(i) + \sum_{i \in \mathcal{W}(\mathbf{x}) \setminus \mathcal{I}(\mathbf{x})} \mathbf{g}(i)\mathbf{d}(i)$$

where $\mathbf{g} = \nabla f(\mathbf{x})$, and since $\sum_{i \in \mathcal{W}(\mathbf{x}) \setminus \mathcal{I}(\mathbf{x})} \mathbf{g}(i)\mathbf{d}(i) > 0$, it follows

$$\sum_{i \in \mathcal{I}(\mathbf{x})} \mathbf{g}(i)\mathbf{d}(i) < 0. \tag{12}$$

Observe that

$$\mathbf{x}[\rho] = \mathbf{x} + \rho \mathbf{d}^{\mathcal{I}(\mathbf{x})},$$

and because of (12) the thesis follows. □

As in the uncostrained case, the convergence of the APGM scheme is ensured by the convergence of the GP iterations (steps 22-26 of Algorithm 2) to a constrained stationary point and by the result established in Proposition 6 . In particular, using similar arguments to [45, Thm. 5.1], we can prove the following proposition.

Proposition 7 *Let x^* be solution to problem (9) and let $\{\mathbf{x}_k\}$ be an infinite sequence generated by Algorithm 2. Then $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^*$.*

Proof First of all, we notice that the line search condition at step 26 of Alg. 2 guarantee that the sequence $\{\mathbf{x}_k\}$ remains in $\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ (see [44, Thm.-Sec. 3]), which is a bounded set due to the strict convexity of the objective function f; then $\{\mathbf{x}_k\}$ is a bounded sequence. If $\mathcal{K}_{\text{GP}}$ is the set of indices corresponding to the iterates generated by the gradient projection method, $\mathcal{K}_{\text{GP}}$ is infinite, the subsequence $\{\mathbf{x}_k : k \in \mathcal{K}_{\text{GP}}\}$ is bounded and it admits at least one limit point. When a monotone version of the Armijo-like condition is considered, Theorem 5.2 in [46] guarantees that every limit point of the subsequence $\{\mathbf{x}_k : k \in \mathcal{K}_{\text{GP}}\}$ is a constrained stationary point for problem (4). On the other hand, for $M > 1$, the same result can be ensured by [18, Thm. 2.4] combined with Proposition 6. In both cases, since f is strictly convex, the solution $\mathbf{x}^*$ is the only stationary point of problem (1). As a consequence, there is a subsequence of $\{\mathbf{x}_k\}$ that converges to $\mathbf{x}^*$ and $\{f(\mathbf{x}_k)\}$ converges to $\{f(\mathbf{x}^*)\}$. In particular, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \nabla f(\mathbf{x}^*)^\top (\mathbf{x}_k - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^\top \nabla^2 f(\mathbf{x}^*)(\mathbf{x}_k - \mathbf{x}^*) \geq \frac{\lambda_{\min}}{2} \|\mathbf{x}_k - \mathbf{x}^*\|^2,$$

where the inequality is guaranteed by the stationarity condition

$$\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \Omega.$$

Since $\{f(\mathbf{x}_k)\}$ converges to $\{f(\mathbf{x}^*)\}$, the thesis follows. □

A complexity result about Algorithm 2 can be obtained for the case $M = 1$, under the assumption of Lipschitz continuity of the gradient of the objective function. Indeed, when ∇f is Lipschitz continuous and a monotone Armijo linesearch is performed at step 26, Theorem 3.2 in [47] is applicable, yielding the convergence rate $\mathcal{O}(1/k)$ on the objective function value. Let observe that ∇f is Lipschitz continous when f is a twice continuously differentiable function over a bounded set. Then, the Lipschitz condition assumption of ∇f on Ω for problem (4) is satisfied.

Proposition 8 *Let x^* be the solution of problem* (9)*, and let $\{\mathbf{x}_k\}$ be the sequence generated by Algorithm 2 with $M = 1$. Then we have*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) = \mathcal{O}\left(\frac{1}{k}\right).$$

Proof Let $\mathbf{x}_{k+1}$ be the point generated from $\mathbf{x}_k$ via the acceleration step [**2D**] (or equiv. [**1D**]), and let denote by $\bar{\mathbf{x}}_{k+1}$ the point generated from $\mathbf{x}_k$ via the monotone Armijo rule at the step 26 of Algorithm 2 (case $M = 1$). By observing that

$$f(\mathbf{x}_{k+1}) \leq f(\bar{\mathbf{x}}_{k+1})$$

the proof of the statement follows from [47, Theorem 3.2]. □

As in the quadratic case, when the prior theoretical result fails to hold, applying only a finite number of acceleration steps suffices to recover the standard convergence behavior of the underlying gradient projection methods.

4 Numerical experiments

We evaluated the performance of the AGM using three gradient step options: BB1, BB2, and $\text{ABB}_{\min}$ [9], comparing them via performance profiles [48, 49]. The numerical tests were designed to assess how the embedded acceleration steps influence the behavior of these gradient methods. To this end, we examined randomly generated QP and BCQP test problems, varying features such as matrix dimensions, condition numbers, and initialization strategies. This experimental setup allows for a comprehensive analysis of the efficiency and robustness of the acceleration strategy across different scenarios.

The experiments were performed in the MATLAB R2024b environment on a 14-inch MacBook Pro equipped with an Apple M3 Pro chip and 18 GB of RAM, running on macOS Sequoia (Version 15.6.1).

4.1 Unconstrained convex QP

We considered test problems of type (1) that were randomly generated using Matlab functions, with dimensions $1000, 5000, 10000$ and condition number $\kappa(A) \in \{10^4, 10^5, 10^6, 10^7\}$. The problems were designed to vary both the features and properties, altering not only the conditioning but also the method used to generate the vector $\mathbf{b}$. The aim was to create variability in the relationship between the linear term and the quadratic term. In the first set of problems, the eigenvalues were designed to follow a uniform bimodal distribution. Specifically, half of the eigenvalues were uniformly sampled within the range $[0, 0.2]$, while the other half were uniformly sampled in $[0.8, 1.0]$. These values were then rescaled to span the interval $[1, \kappa(A)]$. In the second scenario, the eigenvalues were logarithmically distributed over the interval $[1, \kappa(A)]$. This distribution was achieved using the `logspace` function in Matlab, which produces values equally spaced on a logarithmic scale. In the third scenario, the eigenvalues were linearly distributed between 1 and $\kappa(A)$ through the `linspace` function in Matlab, offering a smooth and uniform progression of eigenvalues, thereby representing a distinct structure of the Hessian matrix compared to the previous cases. The algorithm parameters were set to $\gamma = 0.5$, $\text{tol} = 10^{-7}$ and $\text{maxiter} = 8n$. We then considered two different ways to generate the right-hand side vector $\mathbf{b}$ and the solution $\mathbf{x}^*$. In the first way, $\mathbf{b}$ was computed as $\mathbf{b} = A \cdot \mathbf{x}^*$, where $\mathbf{x}^*$ was randomly generated with unit infinity norm. To introduce variability into the problem setup, $\mathbf{x}^*$ was generated in three different ways, using the MATLAB functions `rand`, `randn`, and finally using `sprandn` with a sparsity of 0.4. The initial guess for the solution, $\mathbf{x}_0$, was randomly generated with components in $[-5, 5]$ for each trial. Each test case was run 5 times, with a different random seed for each run to account for randomness in initialization and step size selection. This approach allowed a thorough evaluation of the robustness of the methods under different initial conditions. For the first simulation, we ran a total of 540 problems. In the second strategy, $\mathbf{b}$ was set as a vector of ones and we have a total of 180 problems. The initial guess $\mathbf{x}_0$ was also generated randomly with components in the interval $[-5, 5]$. For BB1, BB2 and ABB_{min} we also considered $\mathbf{x}_0 = \mathbf{0}$ as the starting point (reported as BB1(0), BB2(0) and $\text{ABB}_{\text{min}}(0)$ in the performance profiles).

It is important to note that for quadratic problems where $\|\mathbf{b}\|$ is very small, the results might be misleading. In such cases, only a few acceleration steps — just one if $\|\mathbf{b}\| = 0$ — may lead directly to the solution. In contrast, gradient methods do not gain significant advantages when applied to purely quadratic problems, like those in [41]. For this reason problems with such characteristics were removed before making our performance profiles.

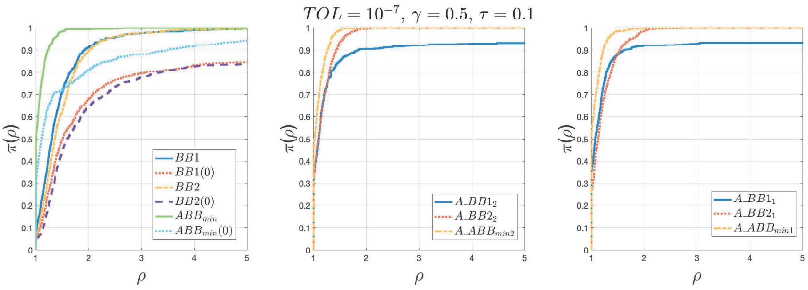

Fig. 6 Comparison of BB1, BB2, ABB_{min}: standard and accelerated versions.

Figure 6 illustrates a comparison of the three selected algorithms across the three versions analyzed. The performance profile on the left shows how ABB_{min} outperforms both BB1 and BB2 algorithms. We also note that using **0** as a starting point does not seem to offer any advantage over random selection. Therefore, we consider only the latter in subsequent comparisons.

We include all three steplength strategies in our numerical experiments, since our goal was not to design a specific algorithm but rather to understand the effect of acceleration steps within different gradient strategies. Notably, the hierarchy among BB1, BB2, and ABB$_{\min}$ shifts when moving to the accelerated versions. In particular, the advantage of ABB$_{\min}$ is strongly reduced in comparison to BB2. The ABB$_{\min}$ strategy avoids stagnation in BB2 algorithm by occasionally incorporating BB1 steps. The performance profiles in Figure 6 suggests that adding acceleration steps to BB2 is sufficient to prevent this problem, making the use of BB1 steps unnecessary, if not counterproductive. Figure 7 illustrates the performance profiles of the algorithms BB1, BB2, and ABB$_{\min}$ in comparison to A_BB1$_1$ and A_BB1$_2$, A_BB2$_1$ and A_BB2$_2$, A_ABB$_{\min 1}$ and A_ABB$_{\min 2}$ respectively. The performance profiles show that acceleration clearly improves the performance of the gradient methods, without showing a significant difference between the use of the [1D] or [2D] steps.

To better understand the effectiveness of our acceleration strategy in relation to the conditioning of the problems, in Figure 8 we compare the scenarios where $\kappa(A) = 10^4$ and $\kappa(A) = 10^7$. The results clearly show that the impact of acceleration is greatest in the most poorly conditioned problems. This supports our conjecture that in ill-conditioned problems far from the solution, the linear and quadratic terms are typically badly scaled. Consequently, the effect of restoring balance between them is more pronounced.

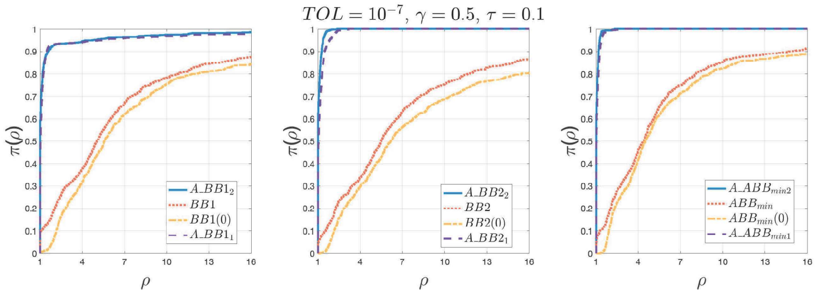

Fig. 7 Standard vs accelerated BB algorithms.

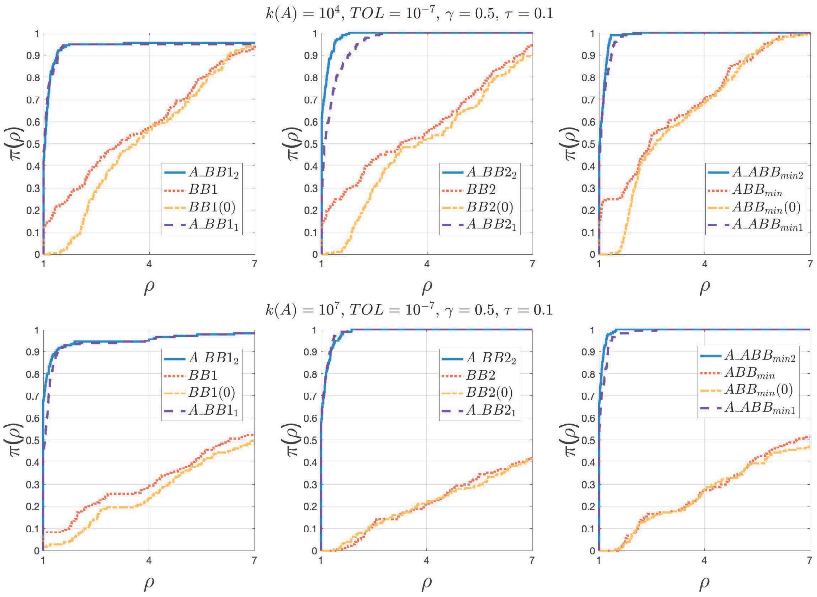

Fig. 8 Performance profiles: problems with $\kappa(A) = 10^4$ and $\kappa(A) = 10^7$.

A comparison between the [1D] and [2D] acceleration steps reveals that the latter is generally more reliable. We conjecture that when the quadratic and linear terms are well balanced, a step focused solely on the quadratic term may not be entirely appropriate. In other words, if the quadratic term $\mathbf{x}_k^\top A\mathbf{x}_k$ is very large with respect to the linear term $\mathbf{b}^\top\mathbf{x}_k$, minimizing in the affine subspace generated by $\mathbf{x}_k$ can be extremely useful; this is what happens at the first iterations when usually the two terms are very unbalanced. Otherwise, minimizing in the subspace generated by $\mathbf{x}_k$ and $\mathbf{g}_k$ may prove to be a safer choice. Therefore, [2D] steps in the AGD framework may give more robustness to the overall strategy.

4.2 Bound constrained convex QP

We analyzed test problems of type (9), which are based on those introduced in [50]. Each problem is defined by the following parameters: the dimension of the problem $n \in \{10000, 20000\}$ and the condition number $\kappa(A) \in \{10^4, 10^5, 10^6\}$. The components of the optimal solution $\mathbf{x}^*$ are sampled from a uniform distribution in $(-1, 1)$ and $\mathbf{b} = A\mathbf{x}^*$. The Hessian matrix A is defined as follows:

$$A = GDG^\top,$$

where D is a diagonal matrix, and G is constructed as

$$G = (I - 2\mathbf{p}_3\mathbf{p}_3^\top)(\mathbf{I} - 2\mathbf{p}_2\mathbf{p}_2^\top)(I - 2\mathbf{p}_1\mathbf{p}_1^\top),$$

with $\mathbf{p}_j,\ j \in \{1, 2, 3\}$ being unit vectors randomly generated.

The diagonal elements of matrix D are specified as follows: for every i, the element d_{ii} is determined by $10^{\frac{i-1}{n-1}(\log_{10}(\kappa(A)))}$. The set $\mathcal{A}^*$ is partitioned into subsets $\mathcal{A}_l^*$ and $\mathcal{A}_u^*$ in the following manner: for each $i \in \mathcal{A}^*$, a random variable $\nu_i \in (0,1)$ is produced. The index i is allocated to A_l^* if $\nu_i < 0.5$, and to A_u^* otherwise. The lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$ are defined as:

$$\mathbf{l}(i) = \begin{cases} -1, & i \notin \mathcal{A}^*, \\ \mathbf{x}^*(i), & i \in \mathcal{A}_l^*, \\ -1, & i \in \mathcal{A}_u^*, \end{cases} \qquad \text{and} \qquad \mathbf{u}(i) = \begin{cases} 1, & i \notin \mathcal{A}^*, \\ 1, & i \in \mathcal{A}_l^*, \\ \mathbf{x}^*(i), & i \in \mathcal{A}_u^*. \end{cases}$$

We thus generated a total of 54 strictly convex constrained problems with nondegenerate solutions. The algorithm parameters were set as follows: $\gamma = 0.5$, $\mu = 100$, tol $= 10^{-7}$, and maxiter $= 8n$. We additionally set $\alpha_{\min} = 10^{-10}$, $\alpha_{\max} = 10^6$, and $M_\alpha = 10$, which are typical values from the literature (see, e.g., [9, 24, 41]).

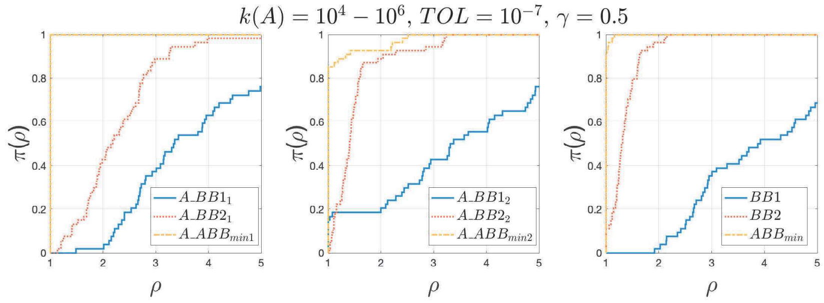

Fig. 9 Comparison of BB1, BB2, ABB_{min}: standard and accelerated versions.

As a preliminary analysis, Figure 11 shows the convergence history for three representative problems with $n = 20000$ variables, corresponding to condition numbers $\kappa(A) = 10^4, 10^5,$ and 10^6. These examples illustrate the overall behavior of the algorithms under consideration. The gap between standard and accelerated algorithms appears less pronounced compared to the unconstrained case, but it remains significant. The computational results for the entire set of test problems are summarized in the performance profiles in Figures 9 and 10. The performance profile on the left of Figure 9 shows that, as in the unconstrained case, ABB_{min} is the best performing algorithm, followed by BB2. In contrast, BB1 is noticeably the least competitive of the three algorithms under consideration. Interestingly, when we move to the accelerated algorithms, the ranking remains consistent among the three, unlike in the unconstrained case.

About the comparison between standard algorithms and their respective accelerated versions, the performance profiles in Figure 10 confirm the effectiveness of the acceleration strategy, particularly for problems with higher condition numbers.

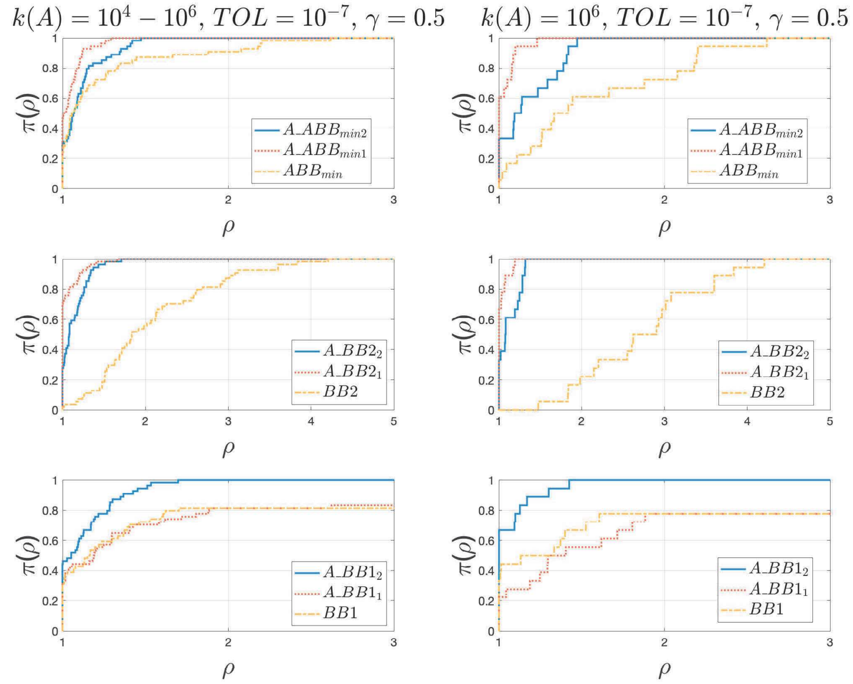

Fig. 10 BCQP problems. BB Gradient methods vs accelerated versions.

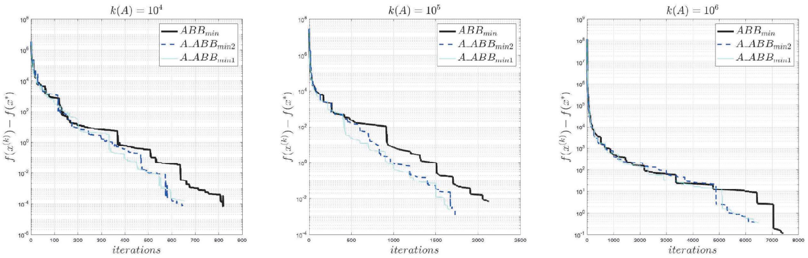

Fig. 11 A_ABB$_{\text{min}}$ vs ABB$_{\text{min}}$: BCQP test problem with 20000 variables. Convergence history.

Although less pronounced than in the unconstrained case, the benefits of using acceleration steps still justify their use. As for the unconstrained case, the option [2D] appears to be more reliable than [1D] for the acceleration steps.

5 General unconstrained minimization

We now discuss the extension of our acceleration strategy to the general unconstrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable objective function. It is not straightforward to directly extend the 2D minimization phase. For a general nonquadratic function, in fact, the 2D minimization subproblem $\min_{\alpha,\beta} f(\mathbf{x}_k - \alpha\mathbf{g}_k - \beta\mathbf{x}_k)$ does not have a closed-form solution. Our aim is to capture the central idea of the 2D acceleration strategy by combining the steepest descent direction $\mathbf{g}_k$ with a second direction $\mathbf{d}_k$ in selected iterations of a generic gradient method. As for the quadratic case, because we want a search direction to be a linear combination of $\mathbf{g}_k$ and $\mathbf{d}_k$, we consider the problem

$$\operatorname{argmin} \tilde{\psi}_k(\alpha,\beta) \equiv f(\mathbf{x}_k - \alpha\tilde{\mathbf{d}}_1 - \beta\tilde{\mathbf{d}}_2) \tag{13}$$

where

$$\tilde{\mathbf{d}}_1 = \frac{\mathbf{g}_k}{\|\mathbf{g}_k\|} \quad \text{and} \quad \tilde{\mathbf{d}}_2 = \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|}. \tag{14}$$

We propose to compute an approximate solution of (13) performing a single line search step from $\mathbf{0} \in \mathbb{R}^2$ along its steepest descent direction. Notice that

$$\nabla\tilde{\psi}_k(\mathbf{0}) = \begin{pmatrix} -\nabla f(\mathbf{x}_k)^\top(\tilde{\mathbf{d}}_1) \\ -\nabla f(\mathbf{x}_k)^\top(\tilde{\mathbf{d}}_2) \end{pmatrix} = \begin{pmatrix} -\|\mathbf{g}_k\| \\ -\frac{\mathbf{g}_k^\top\mathbf{d}_k}{\|\mathbf{d}_k\|} \end{pmatrix}.$$

A steplength $\eta_k > 0$ is computed that satisfies the Wolfe conditions [51] for $\tilde{\psi}_k$, which ensure sufficient decrease and curvature:

$$\begin{aligned} \tilde{\psi}_k\big(\mathbf{0} - \eta_k\nabla\tilde{\psi}_k(\mathbf{0})\big) &\le \tilde{\psi}_k(\mathbf{0}) - c_1\eta_k\,\|\nabla\tilde{\psi}_k(\mathbf{0})\|^2 \\ &= \tilde{\psi}_k(\mathbf{0}) - c_1\eta_k\Big(\|\mathbf{g}_k\|^2 + \Big(\frac{\mathbf{d}_k^\top\mathbf{g}_k}{\|\mathbf{d}_k\|}\Big)^2\Big) \end{aligned} \tag{15}$$

$$\begin{aligned} \nabla\tilde{\psi}_k\big(\mathbf{0} - \eta_k\nabla\tilde{\psi}_k(\mathbf{0})\big)^\top - \nabla\tilde{\psi}_k(\mathbf{0})) &\ge -c_2\,\|\nabla\tilde{\psi}_k(\mathbf{0})\|^2 \\ &= -c_2\Big(\|\mathbf{g}_k\|^2 + \Big(\frac{\mathbf{d}_k^\top\mathbf{g}_k}{\|\mathbf{d}_k\|}\Big)^2\Big). \end{aligned} \tag{16}$$

where $0 < c_1 \le c_2 < 1$ are two constants. It is interesting to note that conditions (15-16) can be reformulated as Wolfe conditions for $f(\mathbf{x})$ as follows:

$$\begin{gathered} f(\mathbf{x}_k - \eta_k\mathbf{p}_k) \le f(\mathbf{x}_k) - c_1\eta_k\mathbf{g}_k^\top\mathbf{p}_k, \\ \nabla f(\mathbf{x}_k - \eta_k\mathbf{p}_k)^\top\mathbf{p}_k \le c_2\mathbf{g}_k^\top\mathbf{p}_k \end{gathered}$$

where

$$\mathbf{p}_k = \mathbf{g}_k + \mathbf{d}_k\frac{\mathbf{d}_k^\top\mathbf{g}_k}{\|\mathbf{d}_k\|^2} = \|\mathbf{g}_k\|\left(\tilde{\mathbf{g}_k} + \tilde{\mathbf{d}}_k\cos\theta_k\right). \tag{17}$$

In order to justify the use of the normalized directions (14), let us consider

$$\hat{\psi}_k = f(\mathbf{x}_k - \alpha\mathbf{g}_k - \beta\mathbf{d}_k) \tag{18}$$

instead of $\tilde{\psi}$. The Wolfe conditions for (18) implie Wolfe conditions for $f(\mathbf{x})$ along:

$$\hat{\mathbf{p}}_k = \|\mathbf{g}_k\|^2\mathbf{g}_k + \mathbf{d}_k\mathbf{g}_k^\top\mathbf{d}_k = \|\mathbf{g}_k\|^2\left(\mathbf{g}_k + \frac{\mathbf{d}_k{}^\top\mathbf{g}_k}{\|\mathbf{g}_k\|^2}\mathbf{d}_k\right). \tag{19}$$

It is straightforward to show that the direction (17) is invariant with respect to the scaling of the objective function $f(\mathbf{x})$, as long as $\mathbf{d}_k$ is also invariant, contrary to what happens for (19). Furthermore, the search direction $\mathbf{p}_k$ satisfies other important theoretical properties. Specifically, the descent direction $-\mathbf{p}_k$ satisfies the direction assumptions required by Dai in [52, eq. (16)-(17)] and by Hager and Zhang (2004) in their nonmonotone line search algorithm (2.4 and 2.5 in [53]):

$$\begin{aligned} -\mathbf{p}_k^\top\mathbf{g}_k &= -\mathbf{g}_k^\top\mathbf{g}_k - \frac{(\mathbf{d}_k^\top\mathbf{g}_k)^2}{\|\mathbf{d}_k\|^2} \leq -\|\mathbf{g}_k\|^2, \\ \mathbf{p}_k^\top\mathbf{p}_k &= \mathbf{g}_k^\top\mathbf{g}_k + 3\frac{(\mathbf{d}_k^\top\mathbf{g}_k)^2}{\|\mathbf{d}_k\|^2} \leq (1+3\zeta)\|\mathbf{g}_k\|^2 \end{aligned}$$

We are now able to present the G_AGM (Generalized AGM) for general nonlinear optimization problems.

Algorithm 3 G_AGM (Generalized AGM)

```
Initialization: $\mathbf{x}_0 \in \mathbb{R}^n$; $k = 0$; $\gamma, \sigma \in (0,1)$; $0 < \alpha_{\min} < \alpha_{\max}$; $M \geq 1$; maxiter; $\mu \in \mathbb{N}$;
$\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
while stop_condition do
    $k = k + 1$
    compute $\mathbf{d}_k$
    if acceleration_condition then
        $\mathbf{p}_k = -(\mathbf{g}_k + \mathbf{d}_k \frac{\mathbf{d}_k^\top\mathbf{g}_k}{\|\mathbf{d}_k\|^2})$
    else
        $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ is computed by any stepsize rule
        $\mathbf{p}_k = -\alpha_k\mathbf{g}_k$
    end if
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \rho_k\mathbf{p}_k$
    where the steplength $\rho_k$ satisfies the generalized Armijo condition [44]:
    $f(\mathbf{x}_k + \rho_k\mathbf{p}_k) \leq \max_{0\leq j\leq \min(k,M-1)} f(\mathbf{x}_{k-j}) + \sigma\rho_k\mathbf{p}_k^\top\mathbf{g}_k$
    $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
end while
```

The R-linear convergence of the scheme can be guaranteed assuming that f is uniformly convex. Indeed, in this case, given that $\mathbf{p}_k$ satisfies conditions (16) and (17) in [52], the result is ensured by [52, Theorem 3.1]. G_AGM provides a very generic scheme, allowing considerable freedom both in the choice of the auxiliary direction $\mathbf{d}_k$ and in the choice of the steplength rule adopted in the gradient method strategy. Another crucial point is to establish an appropriate criterion (`acceleration_condition` in Algorithm 3) for activating the acceleration strategy that does not adversely affect the convergence properties of the underlying gradient method. All these issues will be explored in depth in future work, comparing different choices and strategies. Note that with this choice the search direction $\mathbf{s}_k = \alpha \mathbf{g}_k + \beta \mathbf{d}_k$ can be interpreted as the gradient of the function

$$\alpha f(\mathbf{x}) + \frac{1}{2}\beta \|\mathbf{x}\|^2,$$

which brings us back to the observation made previously for the quadratic case concerning (7).

5.1 Numerical experiments

We present some very preliminary computational results that provide an initial overview of the potential of our approach. Motivated by experiences gained with the algorithm in the quadratic case, in our computational experiments we set

$$\mathbf{d}_k = \mathbf{x}_k.$$

We used the same `acceleration condition` and `stop_condition` as in Algorithm 1 (line 5), with $\text{tol}_g = 10^{-7}$, $\text{tol}_f = 10^{-10}$, $\text{maxiter} = 10^4$. In addition we set a maximum number of acceleration steps equal to 5, and an additional stopping criterion on the stagnation of the objective function:

$$|f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})| \leq \text{tols}_f \, |f(\mathbf{x}_k)|,$$

with $\text{tols}_f = 10^{-4}$. For the steplength selection in the gradient iteration we considered the $\text{ABB}_{\min}$, with parameters setting as in the previous experiments.

Alternatives for $\mathbf{d}_k$ and more effective rules to be used in the `acceleration condition` are currently being studied and will be the subject of a forthcoming work.

First, to illustrate the previously discussed property of scale invariance of (17) with respect to (19), we made some computational experiments with the `tridiagonal2` function $f(\mathbf{x})$ with 500 variables from the CUTEst [54] collection. Namely, we run Algorithm 3 on the function $f^\omega(\mathbf{x}) = \omega f(\mathbf{x})$ for various values of the scaling factor ω. The results shown in Table 1 demonstrate the clear superiority of $\mathbf{p}_k$ vs $\hat{\mathbf{p}}_k$ in terms of stability.

In order to obtain a preliminary but, in our opinion, useful idea of the effectiveness of G_AGM, we run it on a set of 100 unconstrained optimization problems from the CUTEst collection by Andrei [54]. The results are presented in Table 2, which compares G_AGM equipped with $\text{ABB}_{\min}$ rule and the corresponding gradient scheme without acceleration steps. Our analysis excludes instances where the algorithms either

ω	iterations	
	$\mathbf{p}_k$	$\hat{\mathbf{p}}_k$
10^{-3}	31	48
10^{-2}	30	39
10^{-1}	30	48
1	30	84
10^{1}	29	39
10^{2}	31	49
10^{3}	31	34

Table 1 $\mathbf{p}_k$ vs $\hat{\mathbf{p}}_k$ on the `tridiagonal2` problem.

converged to distinct local minimizers (indicated by *) or exceeded the maximum iteration limit without converging (indicated by -). On the remaining 84-problem subset, the accelerated algorithm required fewer iterations in 58.3% of cases. For the remaining instances, it performed worse in 28.6% of cases and equally in the rest. We also note that the test set includes quadratic problems, where the acceleration strategy again demonstrates its efficiency, corroborating the findings from the AGM experiments in Section 4. About the non quadratic problems, the results appear quite promising, although not yet completely consistent. Our initial experiences show that, compared to the quadratic case, the accelerated algorithm appears to be much more sensitive to the choice of parameters. We also believe that an appropriate alternative choice of direction $\mathbf{d}_k$ should be considered very carefully.

6 Conclusion

Gradient methods remain a cornerstone of numerical optimization, warranting the continued exploration of performance-enhancing variations.

The introduction of acceleration steps within gradient methods, presented in this paper, appears as a possible strategy with promising prospects. In particular, for convex quadratic problems, considering the contribution of the linear and quadratic parts of the objective function could be useful in designing line searches in acceleration steps. Numerical experiments on convex QP test problems indicate that BB gradient methods, modified with our acceleration strategy, show improved convergence performance. While our analysis is preliminary and requires further investigation on more general and challenging problems, this work represents a first step in that direction through an initial extension of the accelerated algorithm to general nonlinear minimization.

Motivated by the encouraging numerical results, we are currently working on further developing the approach proposed in this work, with the aim of designing effective methods for nonquadratic problems, including within the broader framework of constrained optimization.

Acknowledgments

This work has greatly benefited from discussions with Valeria Ruggiero and Luca Zanni. Their comments and suggestions allowed us to deepen and focus on many issues, particularly those related to problem (4). The authors would like to thank the anonymous referees for their helpful comments, which helped to improve the manuscript.

Funding

This work was partially supported by the Italian Ministry of University and Research (MIUR) through the PRIN 2022 project “Numerical Optimization with Adaptive Accuracy and Applications to Machine Learning” (Grant No. 2022N3ZNAX) and the PNRR PRIN 2022 project “A multidisciplinary approach to evaluate ecosystem resilience under climate change” (Grant No. P2022WC2ZZ).

References

[1] Dai, Y.H., Fletcher, R.: Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. Numerische Mathematik **100**(1), 21–47 (2005)

[2] Loris, I., Bertero, M., De Mol, C., Zanella, R., Zanni, L.: Accelerating gradient projection methods for ℓ_1-constrained signal recovery by steplength selection rules. Applied and Computational Harmonic Analysis **27**(2), 247–254 (2009)

[3] Serafini, T., Zanghirati, G., Zanni, L.: Gradient projection methods for quadratic programs and applications in training support vector machines. Optimization Methods and Software **20**(2-3), 353–378 (2005)

[4] Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. IEEE Transactions on Signal Processing **57**(7), 2479–2493 (2009)

[5] Zanella, R., Zanghirati, G., Cavicchioli, R., Zanni, L., Boccacci, P., Bertero, M., Vicidomini, G.: Towards real-time image deconvolution: application to confocal and sted microscopy. Scientific Reports **3**(1), 2523 (2013)

[6] Zhu, M., Wright, S.J., Chan, T.F.: Duality-based algorithms for total-variation-regularized image restoration. Computational Optimization and Applications **47**(3), 377–400 (2010)

[7] Akaike, H. Annals of the Institute of Statistical Mathematics **11**(1), 1–16 (1959) https://doi.org/10.1007/BF01831719

[8] Fletcher, R.: A limited memory steepest descent method. Mathematical Programming, Series A **135**, 413–436 (2012)

[9] Frassoldati, G., Zanni, L., Zanghirati, G.: New adaptive stepsize selections in gradient methods. Journal of Industrial and Management Optimization **4**(2), 299–312 (2008)

[10] Zhou, B., Gao, L., Dai, Y.H.: Gradient methods with adaptive step-sizes. Computational Optimization and Applications **35**(1), 69–86 (2006)

[11] Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA Journal of Numerical Analysis **8**(1), 141–148 (1988)

[12] Fletcher, R.: On the Barzilai-Borwein method. In: Optimization and Control with Applications, pp. 235–256. Springer, Boston (2005)

[13] Ferrandi, G., Hochstenbach, M.E., Krejić, N.: A harmonic framework for stepsize selection in gradient methods. Computational Optimization and Applications **85**(1), 75–106 (2023)

[14] Friedlander, A., Martínez, J.M., Molina, B., Raydan, M.: Gradient method with retards and generalizations. SIAM Journal on Numerical Analysis **36**(1), 275–289 (1999)

[15] di Serafino, D., Ruggiero, V., Toraldo, G., Zanni, L.: On the steplength selection in gradient methods for unconstrained optimization. Applied Mathematics and Computation **318**, 176–195 (2018) https://doi.org/10.1016/j.amc.2017.12.019

[16] Bonettini, S., Zanella, R., Zanni, L.: A scaled gradient projection method for constrained image deblurring. Inverse Problems **25**(1), 015002 (2009)

[17] Antonelli, L., De Simone, V., di Serafino, D.: On the application of the spectral projected gradient method in image segmentation. Journal of Mathematical Imaging and Vision **54**(1), 106–116 (2016) https://doi.org/10.1007/s10851-015-0588-3

[18] Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10**, 1196–1211 (2000)

[19] Birgin, E.G., Martínez, J.M., Raydan, M.: Inexact spectral projected gradient methods on convex sets. IMA Journal of Numerical Analysis **23**(4), 539–559 (2003)

[20] Pospíšil, L., Dostál, Z.: The projected Barzilai–Borwein method with fall-back for strictly convex QCQP problems with separable constraints. Mathematics and Computers in Simulation **145**, 79–89 (2018)

[21] Gonzaga, C.C., Schneider, R.M.: On the steepest descent algorithm for quadratic functions. Computational Optimization and Applications **63**, 523–542 (2016)

[22] Nocedal, J., Sartenaer, A., Zhu, C.: On the behavior of the gradient norm in the steepest descent method. Computational Optimization and Applications **22**(1), 5–35 (2002)

[23] Crisci, S., Rebegoldi, S., Toraldo, G., Viola, M.: Barzilai–Borwein-like rules in

proximal gradient schemes for ℓ_1-regularized problems. Optimization Methods and Software **39**(3), 601–633 (2024)

[24] Crisci, S., Ruggiero, V., Zanni, L.: Steplength selection in gradient projection methods for box-constrained quadratic programs. Applied Mathematics and Computation **356**, 312–327 (2019)

[25] Crisci, S., Porta, F., Ruggiero, V., Zanni, L.: Spectral properties of barzilai–Borwein rules in solving singly linearly constrained optimization problems subject to lower and upper bounds. SIAM Journal on Optimization **30**(2), 1300–1326 (2020)

[26] Birman, M.S.: Nekotorye ocenki dlja metoda naiskoreišego spuska. Uspehi Matem. Nauk (N.S.) **5**, 152–155 (1950)

[27] Forsythe, G.E.: On the asymptotic directions of the s-dimensional optimum gradient method. Numerische Mathematik **11**, 57–76 (1968)

[28] Pronzato, L., Wynn, H.P., Zhigljavsky, A.: In: Pronzato, L., Zhigljavsky, A. (eds.) A Dynamical-System Analysis of the Optimum s-Gradient Algorithm, pp. 39–80. Springer, New York, NY (2009)

[29] Powell, M.J.D.: A new algorithm for unconstrained optimization. In: Rosen, J.B., Mangasarian, O.L., Ritter, K. (eds.) Nonlinear Programming, pp. 31–65 (1970)

[30] Bartholomew-Biggs, M.: A Newton method with a two-dimensional line search. Advanced Modeling and Optimization **5**(2), 109–127 (2003)

[31] di Serafino, D., Toraldo, G., Viola, M.: Using gradient directions to get global convergence of Newton-type methods. Applied Mathematics and Computation **409**, 125612 (2021)

[32] Bardsley, J., Vogel, C.: A nonnegatively constrained convex programming method for image reconstruction. SIAM Journal on Scientific Computing **25**(4), 1326–1343 (2004)

[33] Bartholomew-Biggs, M.C., Forbes, A.B.: A two-dimensional search used with a non-linear least squares solver. Journal of Optimization Theory and Applications **104**(1), 181–198 (2000)

[34] Dennis, J.E., Mei, H.H.W.: An unconstrained optimization algorithm which uses function and gradient values. Journal of Optimization Theory and Applications **28**(4), 453–482 (1979)

[35] Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics **4**(5), 1–17 (1964)

[36] Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont, MA, USA

(1999)

[37] Chronopoulos, A.T., Gear, C.W.: s-step iterative methods for symmetric linear systems. Journal of Computational and Applied Mathematics **25**, 153–168 (1989)

[38] Lara, H., Aleixo, R., Oviedo, H.: Delayed weighted gradient method with simultaneous step-sizes for strongly convex optimization. Computational Optimization and Applications **89**(1), 151–182 (2024)

[39] Andreani, R., Oviedo, H., Raydan, M., Secchin, L.D.: An extended delayed weighted gradient algorithm for solving strongly convex optimization problems. Journal of Computational and Applied Mathematics **416**, 114525 (2022) https://doi.org/10.1016/j.cam.2021.114525

[40] Oviedo, H., Andreani, R., Raydan, M.: A family of optimal weighted conjugate-gradient-type methods for strictly convex quadratic minimization. Numerical Algorithms **90**(3), 1225–1252 (2022)

[41] Raydan, M., Svaiter, B.F.: Relaxed steepest descent and Cauchy-Barzilai-Borwein method. Computational Optimization and Applications **21**(2), 155–167 (2002)

[42] di Serafino, D., Hager, W.W., Toraldo, G., Viola, M.: On the stationarity for nonlinear optimization problems with polyhedral constraints. Mathematical Programming **205**, 107–134 (2024)

[43] De Angelis, P.L., Toraldo, G.: On the identification property of a projected gradient method. SIAM Journal on Numerical Analysis **30**(5), 1483–1497 (1993)

[44] Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM Journal on Numerical Analysis **23**, 707–716 (1986)

[45] Moré, J.J., Toraldo, G.: On the solution of large quadratic programming problems with bound constraints. SIAM Journal on Optimization **1**(1), 93–113 (1991)

[46] Calamai, P.H., Moré, J.J.: Projected gradient methods for linearly constrained problems. Mathematical Programming **39**(1), 93–116 (1987)

[47] Bonettini, S., Prato, M.: New convergence results for the scaled gradient projection method. Inverse Problems **31**(9), 095008 (2015)

[48] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming, Series A **91**, 201–213 (2002)

[49] Gould, N., Scott, J.: A note on performance profiles for benchmarking software. ACM Transactions on Mathematical Software **43**(2), 15 (2016)

[50] di Serafino, D., Toraldo, G., Viola, M., Barlow, J.: A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds

on the variables. SIAM Journal on Optimization **28**(4), 2809–2838 (2018)

[51] Wolfe, P.: Convergence conditions for ascent methods. SIAM review **11**(2), 226–235 (1969)

[52] Dai, Y.H.: On the nonmonotone line search. Journal of Optimization Theory and Applications **112**(2), 315–330 (2002)

[53] Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM journal on Optimization **14**(4), 1043–1056 (2004)

[54] Andrei, N.: An unconstrained optimization test functions collection. Advanced Modeling and Optimization **10**(1), 147–161 (2008)

Table 2 Performance comparison for $\mathrm{ABB}_{\min}$ with acceleration (`acc`) and without (`noacc`); $\Delta_{\mathtt{itns}} = \mathtt{itns(noacc)} - \mathtt{itns(acc)}$, " $*$ " $\mathrm{ABB}_{\min}$(`acc`) and $\mathrm{ABB}_{\min}$(`noacc`) went to different solutions, " $-$ " method failed to reach the solution. Quadratic problems are in bold.

	$\Delta_{\mathtt{itns}}$	`iterations`			$\Delta_{\mathtt{itns}}$	`iterations`	
Problem		`acc`	`noacc`	**Problem**		`acc`	`noacc`
APEQUAD	20	169	189	EXTTET	0	8	8
ARGLINB	0	1	1	EXTRID1	-17	31	14
ARGLINC	0	1	1	EXTRID2	0	23	23
ARWHEAD	-5	13	8	EXTRIGONO	13	4	17
BDEXP	-3	23	20	EXWHIHOLST	-53	89	36
BDQRTIC	9	73	82	EXWOOD	454	25	479
BIGGSB1	-216	1861	1645	EXPENALTY	*	3	13
BROTRIDIAG	8	31	39	EXPLIN1	*	7857	-
COSINE	*	60	45	EXPLIN2	-	-	-
CUBE	-132	1190	1058	FLETCHCR	-564	4272	3708
CURLY20	530	1442	1972	FLETCBV3	-	-	-
DIAGONAL1	5	103	108	FULLFH1	-9	243	234
DIAGONAL3	12	83	95	**FULLFH2**	-598	4448	3850
DIAGONAL4	-7	11	4	FULLFH3	1	2	3
DIAGONAL5	-4	9	5	GENHUMPS	-14	1121	1107
DIAGONAL6	-	-	-	GENPSC1	6	13	19
DIAGONAL7	1	5	6	GENROSENBR	-	-	-
DIAGONAL8	1	6	7	GENWHIHOLST	-	-	-
DIAGONAL9	11	95	106	GEQUARTIC	3	9	12
DIAGONAL2	-23	188	165	HAGER	1	28	29
DIXMAANA	0	6	6	HARKERP2	19	73	92
DIXMAANB	1	4	5	HIMMELBG	-3	28	25
DIXMAANC	1	5	6	HIMMELH	-	-	-
DIXMAAND	2	5	7	INDEF	*	36	50
DIXMAANE	8	117	125	LIARWHD	-8	48	40
DIXMAANF	21	79	100	MCCORMCK	-	-	-
DIXMAANG	89	24	113	NONDIA	-9	16	7
DIXMAANH	72	40	112	NONSCOMP	0	29	29
DIXMAANI	129	430	559	NONQUAR	34	113	147
DIXMAANJ	83	108	191	**PERQUAD**	17	180	197
DIXMAANK	92	52	144	**PERQUADIAG**	2	17	19
DIXMAANL	28	71	99	PERTRIDQUAD	20	172	192
DIXON3DQ	43	1813	1856	**POWER**	943	1295	2238
DQDRTIC	-1	14	13	**PPQUADRATIC**	19	21	40
EDENSCH	0	18	18	QF1	12	193	205
EG2	*	165	72	QF2	28	187	215
ENGVAL1	-3	23	20	**QP1**	-7	13	6
EP1	1	3	4	QP2	7	123	130
EPSC1	1	13	14	QUARTC	0	1	1
EXBD1	0	15	15	RAYDAN1	11	87	98
EXBEALE	4	28	32	RAYDAN2	1	6	7
EXCLIFF	*	1	1	SINCOS	1	13	14
EXSCHNB	-7	10	3	SINE	337	871	1208
EXSCHNF	0	15	15	SINQUAD	-6	207	201
EXFREROTH	1	26	27	STAIRCASE1	107	397	504
EXHIEVERT	*	2	11	STAIRCASE2	-85	526	441
EXHIMMBLA	0	14	14	TDIAGONAL1	-2	18	16
EXMARATOS	-50	170	120	TDIAGONAL2	*	63	78
EXPOWELL	13	92	105	**TRIDIA**	44	374	418
EXROSENBR	74	24	98	VARDIM	-	-	-