manuscript No.
(will be inserted by the editor)

Preconditioned Proximal Gradient Methods with Conjugate Momentum: A Subspace Perspective

Jian Chen · Xinmin Yang

Received: date / Accepted: date

Abstract In this paper, we propose a descent method for composite optimization problems with linear operators. Specifically, we first design a structure-exploiting preconditioner tailored to the linear operator so that the resulting preconditioned proximal subproblem admits a closed-form solution through its dual formulation. However, such a structure-driven preconditioner may be poorly aligned with the local curvature of the smooth component, which can lead to slow practical convergence. To address this issue, we develop a subspace proximal Newton framework that incorporates curvature information within a low-dimensional subspace. At each iteration, the search direction is obtained by minimizing a proximal Newton model restricted to a two-dimensional subspace spanned by the current preconditioned proximal gradient direction and a momentum direction derived from the previous iterate. By orthogonalizing the subspace basis with respect to the local Hessian-induced metric, the resulting two-dimensional nonsmooth subproblem can be efficiently approximated by solving two one-dimensional optimization problems. This orthogonalization plays a crucial role: it allows a single pass of alternating one-dimensional updates to provide a good approximation to the original coupled two-dimensional subproblem while keeping the per-iteration computational cost low. We establish global convergence of the proposed method and prove a Q-linear convergence rate under strong convexity. Comparative numerical experiments demonstrate the effectiveness of the proposed algorithm, particularly on high-dimensional and ill-conditioned problems.

Keywords Composite optimization · Preconditioned proximal gradient method · Subspace method · Conjugate momentum

Mathematics Subject Classification (2010) 65K05 · 90C25 · 90C30

J. Chen
College of Mathematics, Sichuan University, Chengdu 610065, China
chenjian_math@163.com
✉X.M. Yang
National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China
xmyang@cqnu.edu.cn

1 Introduction

Composite optimization problems arise in a wide range of applications, including machine learning, signal processing, and data science. A typical formulation is

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $g : \mathbb{R}^n \to \mathbb{R}$ is a proper, convex, and lower semicontinuous function but not necessarily differentiable. Due to their broad applicability, the design of efficient algorithms for such problems has attracted considerable attention.

In recent years, first-order methods have attracted much attention because of their low computational cost and scalability to large-scale problems. Representative approaches include the proximal gradient method [17], the iterative shrinkage-thresholding algorithm (ISTA) [3], and various primaldual methods[4, 9, 15]. These methods only require gradient information and proximal operators, making them attractive for high-dimensional applications. However, it is well known that the performance of first-order methods can deteriorate significantly when the problem is ill-conditioned, which is frequently encountered in practical applications.

To address this issue, second-order methods have been extensively studied. Classical proximal Newton-type methods [13, 24] exploit curvature information to achieve faster local convergence. Nevertheless, the main difficulty of second-order approaches lies in the computational cost of evaluating or approximating Hessian matrices, which becomes prohibitive for large-scale problems. Besides, introducing non-diagonal preconditioning matrices typically destroys the closed-form property of proximal operators. As a consequence, many practical algorithms rely on diagonal preconditioning [19, 22] to preserve computational efficiency. Although such approaches often improve practical performance, it is generally difficult for them to achieve theoretical guarantees comparable to those of vanilla proxima Newton and quasi-Newton methods.

Between first-order and second-order methods lies another important class of techniques: momentum methods. The key idea of momentum methods is to exploit historical information to accelerate convergence. Classical frameworks of momentum methods include the heavy-ball method [20], Nesterov's accelerated gradient method [16], and the nonlinear conjugate gradient methods [7, 8, 10, 21]. Due to their low computational cost and strong empirical performance, momentum-based methods have become widely used in large-scale optimization problems.

Understanding why momentum methods perform well has attracted considerable attention in recent years. For instance, the heavy-ball method and Nesterov acceleration have been interpreted through the lens of inertial dynamical systems [1, 14, 25, 26]. The nonlinear conjugate gradient method, on the other hand, originates from the conjugacy property of the linear conjugate gradient method and its finite termination property on quadratic problems. However, in large-scale or ill-conditioned settings, the conjugacy property may deteriorate, leading to degraded performance.

To mitigate this issue, Yuan and Stoer [27] proposed a subspace algorithm. The main idea is to construct a low-dimensional subspace using historical search directions together with the current gradient direction. Within this subspace, a second-order model is approximated using finite differences, and the optimal solution of the subspace problem is used to determine conjugate parameters. This strategy can be interpreted as a subspace optimality principle. When the basis and the approximation model are properly chosen, the resulting method can achieve fast asymptotic convergence comparable to full-space second-order algorithms [28]. As a result, subspace algorithms have attracted increasing attention. More recently, Lapucci et al. [11] established global convergence results for subspace methods in the nonconvex setting. This strategy has been extended to multi-objective [5] and constrained optimization problems [12].

Given the strong empirical and theoretical performance of subspace algorithms in smooth optimization, a natural question arises:

Can the subspace algorithm of Yuan and Stoer be extended to composite optimization problems?

In this paper, we address this question by studying a more general class of composite optimization problems with linear operators.

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(Ax), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $g : \mathbb{R}^m \to \mathbb{R}$ is a proper, convex, and lower semicontinuous function but not necessarily differentiable, and $A \in \mathbb{R}^{m\times n}$. Models of the form (1) arise in a wide range of applications such as imaging (especially total variation regularization), sparse recovery, and various linearly constrained optimization problems. Extending subspace algorithms to problems of the form (1) presents several fundamental challenges. First, the proximal operator of the composition $g\circ A$ does not admit a closed-form solution, which makes it difficult to construct efficient descent directions and the associated subspace. In particular, the choice of search directions is crucial for capturing useful curvature information and maintaining desirable convergence properties. Second, even when the search is restricted to a low-dimensional subspace, the resulting nonsmooth subproblem may still be difficult to solve efficiently. Designing computational strategies that balance approximation accuracy and computational efficiency therefore becomes a central challenge in extending subspace methods to composite optimization.

To address these challenges, this paper explores efficient strategies for subspace construction and fast algorithms for solving the subspace model. The main contributions are summarized as follows.

- When the linear operator A has full row rank, we exploit the singular value decomposition (SVD) of A and complete the orthogonal eigenvectors of $A^\top A$ to construct a preconditioner P, which enables the preconditioned dual problem to admit a closed-form solution. In the linear case where A is not of full row rank, we remove inactive constraints so that the remaining rows become full rank, allowing the same preconditioning strategy to be applied. We further investigate the applications of this preconditioning strategy, particularly in linearly constrained optimization problems. By exploiting the structure induced by the proposed preconditioner, we construct a closed-form preconditioned projection, which significantly simplifies the projection step that typically arises in projected gradient methods. As a result, the computational difficulty of evaluating projections onto the feasible set is greatly reduced [6].
- The structure-driven preconditioner P may severely misalign with the local curvature of the smooth term f, leading to ill-conditioned Hessian $P^{-1/2}\nabla^2 f(x)P^{-1/2}$ in the transformed problem. To alleviate this issue, motivated by the affine-invariant property of Newton's method, we introduce a subspace second-order model that improves the conditioning of the problem. Specifically, a local second-order approximation is constructed within a low-dimensional subspace using finite differences. To balance approximation accuracy and computational efficiency, we identify two conjugate directions associated with the local curvature within this subspace. Along each direction, a one-dimensional second-order model is constructed and solved efficiently. The resulting search direction consists of a preconditioned descent direction and its conjugate direction, which leads to a preconditioned proximal gradient method with conjugate momentum ($\mathrm{P^2GM\_CM}$).
- Numerical experiments demonstrate that the conjugate direction plays a crucial role in the performance of the proposed algorithm. The resulting method achieves competitive performance on several representative problems, including LASSO problems, linearly constrained optimization problems, and structured ℓ_1 regularization problems.

The rest of this paper is organized as follows. Section 2 introduces the notation and preliminary results that will be used throughout the paper. In Section 3 we develop the preconditioned proximal gradient framework and describe the construction of the structure-exploiting preconditioner. Section 4 discusses several important applications in which the proposed preconditioning strategy leads to closed-form dual updates. In Section 5 we introduce the subspace acceleration mechanism and the conjugate momentum strategy, and derive the associated search directions. Section 6 presents the complete algorithm together with efficient methods for solving the one-dimensional subproblems and establishes the global and linear convergence results. Numerical experiments demonstrating the efficiency of the proposed method are reported in Section 7. Finally, some conclusions are drawn at the end of the paper.

2 Preliminaries

Throughout this paper, the n-dimensional Euclidean space $\mathbb{R}^n$ is equipped with the inner product $\langle\cdot,\cdot\rangle$ and the induced norm $\|\cdot\|$. Denote by $\mathbb{S}^n_{++}$ ($\mathbb{S}^n_+$) the set of symmetric positive (semi-)definite matrices and by $\mathbb{O}^n$ the set of orthogonal matrices in $\mathbb{R}^{n\times n}$. The rank of a matrix is denoted by $\mathcal{R}(\cdot)$. For a differentiable function f, $\nabla f(x)\in\mathbb{R}^n$ and $\nabla^2 f(x)\in\mathbb{R}^{n\times n}$ denote the gradient and the Hessian of f at x, respectively. For a positive definite matrix H, we define the norm

$$\|x\|_H=\sqrt{\langle x,Hx\rangle}.$$

For simplicity, we denote $[n]:=\{1,2,\ldots,n\}$ and define the n-dimensional unit simplex by

$$\varDelta_n:=\left\{x\in\mathbb{R}^n:\sum_{i\in[n]}x_i=1,\ x_i\geq 0\right\}.$$

To avoid ambiguity, we introduce the partial order $\preceq$ ($\prec$) in $\mathbb{R}^n$ as

$$u\preceq(\prec)v\iff v-u\in\mathbb{R}^n_+(\mathbb{R}^n_{++}),$$

and in $\mathbb{S}^n$ as

$$U\preceq(\prec)V\iff V-U\in\mathbb{S}^n_+(\mathbb{S}^n_{++}).$$

For $b\in\mathbb{R}^m$, we denote the interval

$$[-\infty,b]=[-\infty,b_1]\times\cdots\times[-\infty,b_m].$$

For $a\in\mathbb{R}^n$, $r>0$, and $p\in[1,+\infty]$, the ℓ_p ball centered at a with radius r is defined as

$$\mathrm{B}_p[a,r]:=\{x\in\mathbb{R}^n:\|x-a\|_p\leq r\}.$$

For a proper extended real-valued function $h:\mathbb{R}^n\to(-\infty,+\infty]$, its domain is defined as

$$\operatorname{dom}h:=\{x\in\mathbb{R}^n:h(x)<+\infty\}.$$

The subdifferential of h at $x\in\operatorname{dom}h$ is defined by

$$\partial h(x):=\{v\in\mathbb{R}^n:h(y)\geq h(x)+\langle v,y-x\rangle,\ \forall y\in\mathbb{R}^n\}.$$

The convex conjugate of h is defined as

$$h^*(y)=\sup_{x\in\mathbb{R}^n}\{\langle x,y\rangle-h(x)\}.$$

The proximal operator associated with h is defined by

$$\operatorname{prox}_h(x) = \arg\min_{u \in \mathbb{R}^n} \left\{ h(u) + \frac{1}{2}\|u - x\|^2 \right\}.$$

Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. The normal cone of C at $x \in C$ is defined by

$$N_C(x) := \{v \in \mathbb{R}^n : \langle v, y - x\rangle \le 0,\ \forall y \in C\}.$$

The support function of C is defined as

$$\sigma_C(x) := \sup_{y \in C} \langle x, y\rangle, \quad x \in \mathbb{R}^n.$$

The indicator function of C is defined by

$$\delta_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C. \end{cases}$$

For a symmetric positive definite matrix $H \in \mathbb{S}^n_{++}$ and a nonempty closed convex set $C \subseteq \mathbb{R}^n$, the preconditioned projection of x onto C is defined as

$$\Pi_C^H(x) := \arg\min_{y \in C} \|y - x\|_H.$$

It holds that $z = \Pi_C^H(x)$ if and only if

$$H(x - z) \in N_C(z).$$

A differentiable function f is said to be L-smooth if

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

This implies the descent inequality

$$f(y) \le f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2.$$

The function f is μ-strongly convex if

$$f(y) \ge f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2$$

for all $x, y \in \mathbb{R}^n$.

3 Preconditioned proximal gradient method

In this section we develop a preconditioned proximal gradient framework for solving composite optimization problems of the form (1). In general, the proximal mapping of $g \circ A$ does not admit a closed-form expression, which may significantly increase the cost of each proximal gradient step.To address this difficulty, we introduce a preconditioning strategy that transforms the proximal gradient subproblem into a dual problem with a much simpler structure.

For $P \in \mathbb{S}^n_{++}$, consider the following preconditioned proximal gradient subproblem

$$\min_{x\in\mathbb{R}^n} \nabla f(x^k)^\top (x - x^k) + g(Ax) + \frac{1}{2}\|x - x^k\|_P^2. \tag{2}$$

To analyze this problem, we introduce its saddle-point formulation

$$\min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^m} \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}\|x - x^k\|_P^2 + y^\top Ax - g^*(y),$$

where g^* is the convex conjugate of g. By the minimax theorem, the problem can be equivalently written as

$$\max_{y\in\mathbb{R}^m} \min_{x\in\mathbb{R}^n} \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}\|x - x^k\|_P^2 + y^\top Ax - g^*(y).$$

Let $\tilde{x}^k$ denote the minimizer of problem (2). The optimality condition with respect to x yields

$$\tilde{x}^k = x^k - P^{-1}(\nabla f(x^k) + A^\top y^k), \tag{3}$$

where y^k is the optimal solution of the following dual problem:

$$\min_{y\in\mathbb{R}^m} \frac{1}{2} y^\top AP^{-1}A^\top y + g^*(y) - a^{k\top} y, \tag{4}$$

with

$$a^k := Ax^k - AP^{-1}\nabla f(x^k).$$

The main motivation behind the preconditioned proximal gradient method is to simplify the dual subproblem through a proper choice of the preconditioner P. In particular, if P is chosen such that

$$AP^{-1}A^\top = \boldsymbol{I}_m, \tag{5}$$

then the dual problem (4) reduces to

$$\min_{y\in\mathbb{R}^m} \frac{1}{2}\|y\|^2 + g^*(y) - a^{k\top} y,$$

whose solution is simply

$$y^k = \mathrm{prox}_{g^*}(a^k).$$

Therefore, the key question becomes how to construct a suitable preconditioner P that satisfies condition (5) while remaining computationally tractable.

3.1 Selection of P

We now describe a systematic way to construct a preconditioner satisfying condition (5). The construction relies on the SVD of the matrix A.

Let the SVD of A be

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

where $U \in \mathbb{O}^m$ and $V \in \mathbb{O}^n$. Then

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

Based on this decomposition, we select the preconditioner P as

$$P = \begin{cases} A^\top A, & \mathcal{R}(A) = n, \\ A^\top A + V \begin{bmatrix} \mathbf{0}_{m\times m} & \\ & \tilde{P} \end{bmatrix} V^\top, & \mathcal{R}(A) = m < n, \end{cases} \tag{6}$$

where $\tilde{P} \in \mathbb{S}^{n-m}_{++}$. To further reveal the structure of the preconditioner, define

$$M = \begin{cases} A, & \mathcal{R}(A) = n, \\ \begin{bmatrix} A \\ \mathbf{0}_{(n-m)\times n} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{m\times m} & \\ & \tilde{P}^{1/2} \end{bmatrix} V^\top, & \mathcal{R}(A) = m < n. \end{cases} \tag{7}$$

With this construction, the preconditioner can be written as

$$P = M^\top M.$$

Consequently, $P \in \mathbb{S}^n_{++}$ and satisfies condition (5). This choice of P ensures that the dual subproblem admits a closed-form solution, which significantly simplifies the computation of the proximal gradient step.

3.2 Diagonal preconditioning

A main appeal of iteration (2) is that, for suitable choices of P, the subproblem (4) admits a closed-form solution, making each iteration computationally efficient. It is worth noting, however, that such a choice of P is tightly coupled with the linear operator A in order to guarantee $AP^{-1}A^\top = I_m$. While this property greatly simplifies the dual update, it may introduce a potential drawback.

Specifically, since P is primarily designed to accommodate the structure of A, it may be poorly aligned with the local second-order geometry of f. When the spectral structure of A differs substantially from that of the local curvature $\nabla^2 f(x^k)$, the induced metric $\|\cdot\|_P$ may impose an inappropriate scaling across different directions. This mismatch may lead to additional ill-conditioning and thus slow down the convergence of the overall algorithm, even though the subproblem itself remains easy to solve.

To strike a balance between per-iteration computational cost and improved curvature exploration, we propose a diagonal preconditioning strategy. Instead of fixing P, we allow a variable preconditioner $P = P_k$ in (4) such that the quadratic term in the dual subproblem becomes diagonal; equivalently, we enforce that $AP_k^{-1}A^\top$ is a diagonal full-rank matrix. This diagonal structure decouples the dual variables and allows direction-wise scaling to be adjusted independently. As a result, the preconditioner P_k can better capture the local curvature information while preserving the computational simplicity of the update.The remaining question is how to construct such a preconditioner P_k.

From a preconditioning perspective, the matrix $P = M^\top M$ in (2) can be interpreted as inducing a change of variables $y = Mx$. Denote

$$h(y) = f(x) = f(M^{-1}y).$$

We then apply a diagonal preconditioning matrix D_k to the function h in order to better capture its local geometry. The diagonal matrix D_k can be constructed using various diagonal preconditioning strategies, such as the diagonal Barzilai–Borwein method [] or diagonal BFGS updates []. Once D_k is specified, we define the preconditioning matrix as

$$P_k := M^\top D_k M,$$

where

$$D_k = \operatorname{diag}(d_1^k, \dots, d_n^k),$$

with bounded entries satisfying

$$0 < d_l \le d_i^k \le d_u, \quad \forall k,\ i \in [n].$$

With this construction, we obtain

$$AP_k^{-1}A^\top = \begin{cases} D_k^{-1}, & \mathcal{R}(A) = n, \\ D_{k[m\times m]}^{-1}, & \mathcal{R}(A) = m < n. \end{cases} \tag{8}$$

4 Application

In this section we illustrate how the proposed preconditioning strategy can significantly simplify the computation of the dual subproblem for several important classes of composite optimization problems. By exploiting the structure of the linear operator A, the preconditioner transforms the dual problem into a form where the optimal solution can be obtained via simple projections onto standard convex sets. Consequently, computationally expensive proximal evaluations are replaced by closed-form projection operations.

4.1 Ellipsoidal constrained problems

Consider the ellipsoidal constrained optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x^\top Bx \le b, \end{aligned}$$

where $B \in \mathbb{S}_+^n$ and $\mathcal{R}(B) = m$. Let A be a matrix satisfying $A^\top A = B$. Then the constraint can be written in the form

$$g(Ax) = \delta_{\mathrm{B}_2[\mathbf{0}_m, \sqrt{b}]}(Ax),$$

where $\mathrm{B}_2[\mathbf{0}_m, \sqrt{b}]$ denotes the Euclidean ball with radius $\sqrt{b}$. The conjugate function of g is

$$g^*(y) = \delta_{\mathrm{B}_2[\mathbf{0}_m, 1/\sqrt{b}]}(y) = \begin{cases} 0, & y \in \mathrm{B}_2[\mathbf{0}_m, 1/\sqrt{b}], \\ +\infty, & \text{otherwise}. \end{cases} \tag{9}$$

Applying the preconditioner $P_k = M^\top D_k M$, the resulting dual problem becomes

$$\min_{y \in \mathrm{B}_2[\mathbf{0}_m, 1/\sqrt{b}]} \frac{1}{2} \|y\|^2_{D^{-1}_{k[m\times m]}} - a^{k\top} y.$$

In particular, when $D_k := L_k \boldsymbol{I}_m$, the optimality condition yields the closed-form solution

$$y^k = \Pi_{\mathrm{B}_2[\mathbf{0}_m, 1/\sqrt{b}]}(L_k a^k).$$

Therefore, the dual update reduces to a simple projection onto a Euclidean ball, which can be computed efficiently in closed form.

4.2 Structured ℓ_1 regularization problems

Next we consider structured ℓ_1 regularization problem:

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda \|Ax\|_1,$$

where $\lambda > 0$, $A \in \mathbb{R}^{m\times n}$ and $\mathcal{R}(A) = m$. In this case

$$g(Ax) = \lambda \|Ax\|_1.$$

The conjugate function is

$$g^*(y) = \delta_{\mathrm{B}_\infty[\mathbf{0}_m, \lambda]}(y) = \begin{cases} 0, & y \in \mathrm{B}_\infty[\mathbf{0}_m, \lambda], \\ +\infty, & \text{otherwise}. \end{cases} \tag{10}$$

Using the preconditioner $P_k = M^\top D_k M$, the dual problem becomes

$$\min_{y \in \mathrm{B}_\infty[\mathbf{0}_m, \lambda]} \frac{1}{2} \|y\|^2_{D^{-1}_{k[m\times m]}} - a^{k\top} y.$$

The optimality condition leads to the explicit solution

$$y^k = \Pi_{\mathrm{B}_\infty[\mathbf{0}_m, \lambda]}(D_{k[m\times m]} a^k).$$

Hence the dual update reduces to a projection onto an ℓ_∞ ball, which is equivalent to a simple componentwise clipping operation.

4.3 linear constrained optimization problems

When the linear operator A is not of full row rank, the preconditioning matrix can be constructed using the linearly independent components of A. This observation is particularly useful for linear constrained optimization problems, where redundant or inactive constraints can be removed so that the remaining constraint matrix becomes full row rank. Let us consider the linear constrained optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Bx \preceq b, \\ & Cx = c, \end{aligned}$$

where $B \in \mathbb{R}^{p\times n}$ and $C \in \mathbb{R}^{q\times n}$. At iteration k, we define the working constraint matrix

$$A_k = \begin{bmatrix} B_k \\ C \end{bmatrix},$$

where B_k consists of the selected inequality constraints. The corresponding nonsmooth term is

$$g_k(A_k x) = \delta_{[-\infty, b_k]\times\{c\}}(A_k x).$$

The conjugate function of g_k is given by

$$g_k^*(y) = \sigma_{[-\infty, b_k]\times\{c\}}(y) = \begin{cases} b_k^\top y_{[1:p_k]} + c^\top y_{[p_k+1, p_k+q]}, & y_{[1:p_k]} \succeq \mathbf{0}_{p_k}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{11}$$

Using the preconditioner $P_k = M_k^\top D_k M_k$, the dual problem becomes:

$$\min_{y_{[1:p_k]}\succeq \mathbf{0}_{p_k}} \frac{1}{2}\|y\|^2_{D^{-1}_{k[m\times m]}} + b_k^\top y_{[1:p_k]} + c^\top y_{[p_k+1, p_k+q]} - a^{k\top} y.$$

The key ingredient for applying the proposed framework to linear constrained optimization problems lies in identifying the working matrix A_k. In contrast to classical active-set methods, which attempt to precisely identify the active constraints, our approach only requires excluding several clearly inactive constraints. In many practical situations, removing inactive indices is significantly easier than accurately detecting the full active set.

4.3.1 Simplex constrained optimization problems

Consider the simplex constrained optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \mathbf{0}_n \preceq x, \\ & \mathbf{1}_n^\top x = 1. \end{aligned}$$

In this case, A_k is constructed from the identity matrix by replacing the row corresponding to the active dual index i_k with the vector $\mathbf{1}_n^\top$. More precisely,

$$A_k = -\boldsymbol{I}_n + e_{i_k}(\mathbf{1}_n + e_{i_k})^\top,$$

and

$$g_k(A_k x) = \delta_{[-\infty, \mathbf{0}_{i_k-1}]\times\{1\}\times[-\infty, \mathbf{0}_{n-i_k}]}(A_k x).$$

The conjugate function is

$$g_k^*(y) = \sigma_{[-\infty, \mathbf{0}_{i_k-1}]\times\{1\}\times[-\infty, \mathbf{0}_{n-i_k}]}(y) = \begin{cases} y_{i_k}, & y_{[1:i_k-1, i_k+1:n]} \succeq \mathbf{0}_{n-1}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{12}$$

Since A_k is a rank-one update of of the identity matrix, the Sherman–Morrison formula yields the closed-form expression

$$A_k^{-1} = -\left(\boldsymbol{I}_n - e_{i_k}(\mathbf{1}_n + e_{i_k})^\top\right)^{-1} = -\left(\boldsymbol{I}_n - \frac{-e_{i_k}(\mathbf{1}_n + e_{i_k})^\top}{1 - (\mathbf{1}_n + e_{i_k})^\top e_{i_k}}\right) = -\boldsymbol{I}_n + e_{i_k}(\mathbf{1}_n + e_{i_k})^\top.$$

Using the preconditioner $P_k = A_k^\top D_k A_k$, the dual problem becomes

$$\min_{y_{[1:i_k-1,i_k+1:n]}\succeq \mathbf{0}_{n-1}} \frac{1}{2}\|y\|^2_{D_k^{-1}} + y_{i_k} - a^{k\top} y.$$

From the optimality conditions we obtain

$$y^k = D_k\left(\left[a^k_{[1:i_k-1]}\right]_+ \times \{a^k_{i_k} - 1\} \times \left[a^k_{[i_k+1:n]}\right]_+\right).$$

4.3.2 Capped simplex constranined optimization problems

Next, consider the capped simplex constrained optimization problem with parameter $s > 0$:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \mathbf{0}_n \preceq x \preceq \mathbf{1}_n, \\ & \mathbf{1}_n^\top x \le s. \end{aligned}$$

In this case, A_k is constructed from the identity matrix by replacing the row corresponding to the active dual index i_k with the vector $\mathbf{1}_n^\top$. More precisely,

$$A_k = \boldsymbol{I}_n + e_{i_k}(\mathbf{1}_n - e_{i_k})^\top,$$

and

$$g_k(A_k x) = \delta_{[\mathbf{0}_{k-1},\mathbf{1}_{i_k-1}]\times[-\infty,s]\times[\mathbf{0}_{n-i_k},\mathbf{1}_{n-i_k}]}(A_k x).$$

The conjugate function is

$$g_k^*(y) = \sigma_{[\mathbf{0}_{i_k-1},\mathbf{1}_{i_k-1}]\times[-\infty,s]\times[\mathbf{0}_{n-i_k},\mathbf{1}_{n-i_k}]}(y) = \begin{cases} \sum_{i\in[1:i_k-1,i_k+1:n]} [y_i]_+ + s y_{i_k}, & y_{i_k} \ge 0, \\ +\infty, & \text{otherwise}. \end{cases} \tag{13}$$

Again, since A_k is a rank-one update of the identity matrix, the Sherman–Morrison formula gives

$$A_k^{-1} = \left(\boldsymbol{I}_n + e_{i_k}(\mathbf{1}_n - e_{i_k})^\top\right)^{-1} = \boldsymbol{I}_n - \frac{e_{i_k}(\mathbf{1}_n - e_{i_k})^\top}{1 + (\mathbf{1}_n - e_{i_k})^\top e_{i_k}} = \boldsymbol{I}_n - e_{i_k}(\mathbf{1}_n - e_{i_k})^\top.$$

Applying the preconditioner $P_k = A_k^\top D_k A_k$, the dual problem becomes

$$\min_{y_{i_k}\ge 0} \frac{1}{2}\|y\|^2_{D_k^{-1}} + \sum_{i\in[1:i_k-1,i_k+1:n]} [y_i]_+ + s y_{i_k} - a^{k\top} y.$$

From the optimality conditions we obtain, for $i \ne i_k$,

$$0 \in \frac{y_i^k}{d_i^k} - a_i^k + \begin{cases} 1, & y_i^k > 0, \\ [0,1], & y_i^k = 0, \\ 0, & y_i^k < 0, \end{cases}$$

and

$$0 \in \left(\frac{y_{i_k}^k}{d_{i_k}^k} + s - a_{i_k}^k\right) + N_{[0,+\infty]}(y_{i_k}^k).$$

Since $d_i^k > 0$, we have

$$y_i^k = \begin{cases} d_i^k(a_i^k - 1), & a_i^k > 1, \\ 0, & a_i^k \in [0,1], \\ d_i^k a_i^k, & a_i^k < 0. \end{cases}$$

Therefore,

$$y^k = D_k\Big(a^k_{[1:i_k-1]} - \max\{\min\{a^k_{[1:i_k-1]}, \mathbf{1}_{i_k-1}\}, \mathbf{0}_{i_k-1}\} \\ \times \left[a^k_{i_k} - s\right]_+ \times a^k_{[i_k+1:n]} - \max\{\min\{a^k_{[i_k+1:n]}, \mathbf{1}_{n-i_k}\}, \mathbf{0}_{n-i_k}\}\Big).$$

5 Subspace acceleration and conjugate momentum

Although the diagonal preconditioning strategy improves the scaling of the proximal gradient step and preserves the closed-form structure of the dual subproblem, its capability remains limited. In particular, diagonal preconditioning only performs coordinate-wise scaling and therefore cannot fully capture the coupling between variables or the richer curvature information of the objective function. As a consequence, the practical acceleration obtained from diagonal scaling alone may still be insufficient, especially for ill-conditioned problems.

To further enhance the performance of the algorithm, we incorporate a subspace acceleration mechanism. Within this framework, two key questions naturally arise. The first concerns how to construct a subspace that effectively captures useful curvature and descent information from past iterates. The second concerns how to design an efficient subspace model so that the resulting subproblem can be solved rapidly while maintaining good approximation quality. These issues will be addressed in the following subsections.

5.1 Selection of subspace and approximate model

To exploit historical information while keeping the computational cost low, we construct a low-dimensional subspace that captures useful search directions from recent iterations. In particular, for $k > 1$ we define the two-dimensional subspace

$$\mathcal{L}_k = \mathtt{span}\{v_k, s_k\},$$

where v_k represents the current preconditioned proximal gradient direction and s_k incorporates information from the previous step. Specifically, the direction s_k is defined as

$$s_k := \begin{cases} d_{k-1}, & x^k + d_{k-1} \in \mathrm{dom}(g \circ A), \\ \Pi^{P_k}_{\mathrm{dom}(g\circ A)}(x^k + d_{k-1}) - x^k, & \text{otherwise}. \end{cases} \tag{14}$$

Remark 1 If $\mathrm{dom}(g \circ A) = \mathbb{R}^n$, then $s_k = d_{k-1}$. When $\mathrm{dom}(g \circ A) \neq \mathbb{R}^n$, as in constrained optimization problems, the direction d_{k-1} may become infeasible at the point x^k. In this case, if $P_k = \boldsymbol{I}_n$, then s_k reduces to $\hat{s}_k$ in [12, Eq. (17)]. However, computing the Euclidean projection $\Pi_{\mathrm{dom}(g\circ A)}(x^k + d_{k-1})$ can be expensive. Instead, when d_{k-1} is infeasible we set

$$s_k = \Pi^{P_k}_{\mathrm{dom}(g\circ A)}(x^k + d_{k-1}) - x^k,$$

under our preconditioning framework. Note that even if d_{k-1} is feasible, we may still have $\Pi^{P_k}_{\mathrm{dom}(g\circ A)}(x^k + d_{k-1}) - x^k \neq d_{k-1}$, which explains why the two cases in (14) are distinguished.

The choice of the direction v_k is also crucial. Here we define v_k as the preconditioned proximal gradient direction at x^k, which is obtained by solving the following subproblem:

$$\min_{v\in\mathbb{R}^n} \nabla f(x^k)^\top v + g(Ax^k + Av) + \frac{1}{2}\|v\|_{P_k}^2. \tag{15}$$

Having constructed the subspace $\mathcal{L}_k$, we next define the corresponding subspace model used to refine the search direction. Restricting the step to $\mathcal{L}_k$, we consider the following subspace proximal Newton subproblem:

$$\min_{d\in\mathcal{L}_k} \nabla f(x^k)^\top d + g(Ax^k + Ad) + \frac{1}{2}\|d\|_{H_k}^2, \tag{16}$$

where $H_k := \nabla^2 f(x^k)$. Since $\mathcal{L}_k = \operatorname{span}\{v_k, s_k\}$ is two-dimensional, any $d \in \mathcal{L}_k$ can be written as $d = G_k\alpha$, where $G_k = [v_k, s_k]$ and $\alpha \in \mathbb{R}^2$. Substituting this representation into (16) yields the equivalent two-dimensional optimization problem

$$\min_{\alpha\in\mathbb{R}^2} c_k^\top \alpha + g(Ax^k + AG_k\alpha) + \frac{1}{2}\|\alpha\|_{Q_k}^2, \tag{17}$$

where $c_k = \begin{bmatrix}\nabla f(x^k)^\top v_k \\ \nabla f(x^k)^\top s_k\end{bmatrix}$, $Q_k = \begin{bmatrix} v_k^\top H_k v_k & v_k^\top H_k s_k \\ v_k^\top H_k s_k & s_k^\top H_k s_k\end{bmatrix}$, and $G_k = [v_k, s_k]$.

5.2 Conjugate basis of subspace

Although problem (17) is only two-dimensional, obtaining its exact solution may still be nontrivial due to the presence of the nonsmooth term $g(Ax^k + AG_k\alpha)$. In many practical situations, computing the exact minimizer is unnecessary and may introduce additional computational overhead. Therefore, instead of solving (17) exactly, we aim to construct an efficient approximation of the minimizer.

To this end, we exploit the structure of the subspace model and perform optimization along carefully chosen directions. In particular, by transforming the basis of the subspace into a conjugate basis with respect to the H_k-inner product, the quadratic term becomes diagonal, which enables efficient alternating one-dimensional optimization.

Specifically, we orthogonalize s_k with respect to v_k under the H_k-inner product and define

$$\tilde{s}_k = s_k - \frac{s_k^\top H_k v_k}{v_k^\top H_k v_k} v_k. \tag{18}$$

With this construction we have $v_k^\top H_k \tilde{s}_k = 0$. Consequently, problem (17) can be rewritten in the equivalent form

$$\min_{\alpha\in\mathbb{R}^2} \langle \tilde{c}_k, \alpha\rangle + g(Ax^k + A\tilde{G}_k\alpha) + \frac{1}{2}\|\alpha\|_{\tilde{Q}_k}^2, \tag{19}$$

where $\tilde{c}_k = \begin{bmatrix}\nabla f(x^k)^\top v_k \\ \nabla f(x^k)^\top \tilde{s}_k\end{bmatrix}$, $\tilde{Q}_k = \begin{bmatrix} v_k^\top H_k v_k & 0 \\ 0 & \tilde{s}_k^\top H_k \tilde{s}_k\end{bmatrix}$, and $G_k = [v_k, \tilde{s}_k]$. Instead of solving (19) directly, we adopt an alternating optimization strategy by solving two one-dimensional subproblems. The first subproblem optimizes along the direction v_k:

$$\min_{\alpha_1\in\mathbb{R}} \nabla f(x^k)^\top v_k\alpha_1 + g(Ax^k + \alpha_1 Av_k) + \frac{1}{2} v_k^\top H_k v_k (\alpha_1)^2. \tag{20}$$

The second subproblem optimizes along the orthogonalized direction $\tilde{s}_k$:

$$\min_{\alpha_2 \in \mathbb{R}} \nabla f(x^k)^\top \tilde{s}_k \alpha_2 + g(Ax^k + \alpha_2 A\tilde{s}_k) + \frac{1}{2}\tilde{s}_k^\top H_k \tilde{s}_k (\alpha_2)^2. \tag{21}$$

Let α_1^k and α_2^k denote the minimizers of (20) and (21), respectively. We then define the intermediate direction

$$\tilde{d}_k = \alpha_1^k v_k + \alpha_2^k \tilde{s}_k.$$

To further refine the search direction, we perform an additional line search along $\tilde{d}_k$ by solving the one-dimensional subproblem:

$$\min_{\alpha_3 \in \mathbb{R}} \nabla f(x^k)^\top \tilde{d}_k \alpha_3 + g(Ax^k + \alpha_3 A\tilde{d}_k) + \frac{1}{2}\tilde{d}_k^\top H_k \tilde{d}_k (\alpha_3)^2. \tag{22}$$

Let α_3^k be the minimizer of (22). The final search direction is then given by

$$d_k = \alpha_3^k \tilde{d}_k.$$

Note that $v_k^\top H_k \tilde{s}_k = 0$ and $\tilde{d}_k = \alpha_1^k v_k + \alpha_2^k \tilde{s}_k$, then

$$\tilde{d}_k^\top H_k \tilde{d}_k = \left\|\alpha_1^k v_k\right\|_{H_k}^2 + \left\|\alpha_2^k \tilde{s}_k\right\|_{H_k}^2.$$

Remark 2 We adopt d_k as the final search direction rather than $\tilde{d}_k$ for two reasons. First, in constrained problems the additional scaling step ensures that the resulting direction remains feasible. Second, solving (22) provides an adaptive stepsize along $\tilde{d}_k$, which improves the stability and effectiveness of the search direction.

It remains to compute the Hessian–vector products $H_k v_k$ and $H_k \tilde{s}_k$. To avoid explicitly forming the Hessian matrix, we approximate the Hessian–vector products using finite differences of gradients. Specifically, we use

$$H_k v \approx H_k(v) := \frac{1}{\epsilon}(\nabla f(x^k + \epsilon v) - \nabla f(x^k)),\ v \in \{v_k, s_k\}. \tag{23}$$

Based on this approximation, the one-dimensional subproblems (20) and (21) can be reformulated as

$$\min_{\alpha_1 \in \mathbb{R}} \nabla f(x^k)^\top v_k \alpha_1 + g(Ax^k + \alpha_1 A v_k) + \frac{q_k(v_k)}{2} \left\|v_k\right\|^2 (\alpha_1)^2, \tag{24}$$

and

$$\min_{\alpha_2 \in \mathbb{R}} \nabla f(x^k)^\top \tilde{s}_k \alpha_2 + g(Ax^k + \alpha_2 A\tilde{s}_k) + \frac{q_k(\tilde{s}_k)}{2} \left\|\tilde{s}_k\right\|^2 (\alpha_2)^2, \tag{25}$$

where $q_k(v) \approx v^\top H_k(v),\ v \in \{v_k, \tilde{s}_k\}$. Denote

$$\tilde{d}_k = \alpha_1^k v_k + \alpha_2^k \tilde{s}_k,$$

where α_1^k and α_2^k are the minimizers of (24) and (25), respectively. To further refine the search direction, we solve an additional one-dimensional subproblem along $\tilde{d}_k$:

$$\min_{\alpha_3 \in \mathbb{R}} \nabla f(x^k)^\top \tilde{d}_k \alpha_3 + g(Ax^k + \alpha_3 A\tilde{d}_k) + \frac{q_{k,d}}{2}(\alpha_3)^2, \tag{26}$$

where

$$q_{k,d} := q_k(v_k) \left\|\alpha_1^k v_k\right\|^2 + q_k(\tilde{s}_k) \left\|\alpha_2^k \tilde{s}_k\right\|^2 \tag{27}$$

The final search direction is then defined as

$$d_k := \alpha_3^k \tilde{d}_k, \tag{28}$$

where α_3^k is the minimizer of (26).

The remaining question is whether the obtained search direction d_k satisfies the sufficient descent condition. Before analyzing this property, we first establish the following auxiliary lemma.

Lemma 1 *Let $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper convex and lower semicontinuous function, which is not necessarily differentiable. Assume that x^* is the minimizer of*

$$\min_{x \in \mathbb{R}^n} a^\top x + h(x) + \frac{1}{2} \|x\|_P^2, \tag{29}$$

where $P \succ 0$. Then

$$a^\top x^* + h(x^*) - h(0) \leq - \|x^*\|_P^2 \tag{30}$$

Proof Since $P \succ 0$ and h is convex, the optimality condition of (29) gives

$$0 \in a + Px^* + \partial h(x^*).$$

Combining this with the convexity of h yields

$$h(0) - h(x^*) \geq (-a - Px^*)^\top (0 - x^*).$$

Rearranging the terms gives the desired result.

We are now ready to provide a sufficient condition under which the direction d_k is a descent direction.

Proposition 1 *Assume that there exist constants $0 < c_1 \leq c_2$ and $c_3 > 0$ such that $c_1 \leq q_k(v_k) \leq c_2$, $q_k(\tilde{s}_k) \geq c_1$ and $P_k \succeq c_3 \boldsymbol{I}_n$ in (24), (25) and (15) for all k. Then, the search direction d_k defined in (28) satisfies the following conditions:*

$$\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k) \leq -\frac{c_1}{2} \|d_k\|^2. \tag{31}$$

$$\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k) \leq -\frac{c_3 \min\{1, c_3/c_2\}}{4} \|v_k\|^2. \tag{32}$$

Proof By Lemma 1 and the definition of d_k, we obtain

$$\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k) \leq -q_{k,d} \cdot (\alpha_3^k)^2.$$

On the other hand, we use the fact $\|a\|^2 + \|b\|^2 \geq 1/2 \|a + b\|^2$ together with the definition of $q_{k,d}$, we obtain

$$q_{k,d} \cdot (\alpha_3)^2 = q_k(v_k) \left\|\alpha_3^k \alpha_1^k v_k\right\|^2 + q_k(\tilde{s}_k) \left\|\alpha_3^k \alpha_2^k \tilde{s}_k\right\|^2 \geq \frac{c_1}{2} \left\|\alpha_3^k(\alpha_1^k v_k + \alpha_2^k \tilde{s}_k)\right\|^2 = \frac{c_1}{2} \|d_k\|^2.$$

Therefore,

$$\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k) \leq -\frac{c_1}{2} \|d_k\|^2,$$

which proves (31).

Next we prove (32). Since α_3^k and α_1^k are the minimizers of (26) and (24), respectively, we have

$$
\begin{aligned}
& \nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k) \\
\overset{(\text{set } \alpha_3=\frac{1}{2} \text{ in } (26))}{\leq} & -\frac{q_{k,d}}{2}(\alpha_3^k)^2 + \frac{1}{2}\nabla f(x^k)^\top \tilde{d}_k + g(Ax^k + \frac{1}{2}A\tilde{d}_k) - g(Ax^k) + \frac{q_{k,d}}{8} \\
\overset{(\text{convexity of } g)}{\leq} & \frac{1}{2}\nabla f(x^k)^\top(\alpha_1^k v_k + \alpha_2^k \tilde{s}_k) + \frac{1}{2}g(Ax^k + \alpha_1^k A v_k) + \frac{1}{2}g(Ax^k + \alpha_2^k A\tilde{s}_k) \\
& - g(Ax^k) + \frac{1}{8}(q_k(v_k)\left\|\alpha_1^k v_k\right\|^2 + q_k(\tilde{s}_k)\left\|\alpha_2^k \tilde{s}_k\right\|^2) \\
= & \frac{1}{2}(\nabla f(x^k)^\top v_k \alpha_1^k + g(Ax^k + \alpha_1^k A v_k) - g(Ax^k) + \frac{1}{4}q_k(v_k)\left\|\alpha_1^k v_k\right\|^2) \\
& + \underbrace{\frac{1}{2}(\nabla f(x^k)^\top \tilde{s}_k \alpha_2^k + g(Ax^k + \alpha_2^k A\tilde{s}_k) - g(Ax^k) + \frac{1}{4}q_k(\tilde{s}_k)\left\|\alpha_2^k \tilde{s}_k\right\|^2)}_{\leq 0 \text{ (by Lemma 1)}} \\
\leq & \frac{1}{2}(\nabla f(x^k)^\top v_k \alpha_1^k + g(Ax^k + \alpha_1^k A v_k) - g(Ax^k) + \frac{1}{2}q_k(v_k)\left\|\alpha_1^k v_k\right\|^2) \\
\overset{(\text{any } 0\leq\alpha_1\leq 1)}{\leq} & \frac{1}{2}(\nabla f(x^k)^\top v_k \alpha_1 + g(Ax^k + \alpha_1 A v_k) - g(Ax^k) + \frac{1}{2}q_k(v_k)\left\|v_k\right\|^2(\alpha_1)^2) \\
\overset{(\text{convexity of } g)}{\leq} & \frac{1}{2}(\nabla f(x^k)^\top v_k \alpha_1 + \alpha_1(g(Ax^k + Av_k) - g(Ax^k)) + \frac{c_2}{2c_3}\left\|v_k\right\|_{P_k}^2(\alpha_1)^2) \\
= & \frac{\alpha_1}{2}(\nabla f(x^k)^\top v_k + (g(Ax^k + Av_k) - g(Ax^k)) + \frac{c_2\alpha_1}{2c_3}\left\|v_k\right\|_{P_k}^2) \\
\overset{(\text{any } 0\leq\alpha_1\leq\min\{1,c_3/c_2\})}{\leq} & \frac{\alpha_1}{2}(\nabla f(x^k)^\top v_k + (g(Ax^k + Av_k) - g(Ax^k)) + \frac{1}{2}\left\|v_k\right\|_{P_k}^2) \\
\overset{(\text{by Lemma 1})}{\leq} & -\frac{\alpha_1}{4}\left\|v_k\right\|_{P_k}^2 \\
\leq & -\frac{c_3\min\{1, c_3/c_2\}}{4}\left\|v_k\right\|^2,
\end{aligned}
$$

where the last inequality follows by $P_k \succeq c_3 I$ and the arbitrary of $\alpha_1 \in \big[0, \min\{1, c_3/c_2\}\big]$. The proof is completed.

6 Preconditioned proximal gradient method with cojugate momentum

To guarantee the sufficient condition stated in Proposition 1, for $v \in \{v_k, \tilde{s}_k\}$ we define

$$
q_k(v) := \begin{cases} \max\left\{c_1, \min\left\{\dfrac{v^\top H_k(v)}{\|v\|^2}, c_2\right\}\right\}, & v^\top H_k(v) > 0, \\ \max\left\{c_1, \min\left\{\dfrac{\|H_k(v)\|}{\|v\|}, c_2\right\}\right\}, & v^\top H_k(v) < 0, \\ c_1, & v^\top H_k(v) = 0, \end{cases} \tag{33}
$$

where $H_k(v)$ is defined in (23), c_1 and c_2 are the positive constants introduced in Proposition 1.

The complete preconditioned proximal gradient method with conjugate momentum is described as follows.

Algorithm 1: Preconditioned proximal gradient method with conjugate momentum (P^{2}GM_CM)

Require: $x^0 \in \mathrm{dom} g^A$, $0 < c_1 \leq c_2$, $0 < c_3 \leq c_4$, $\sigma, \gamma \in (0, 1)$

1: **for** $k = 0, \cdots$ **do**
2: Update $c_3 \boldsymbol{I}_n \preceq P_k \preceq c_4 \boldsymbol{I}_n$
3: Compute v_k as the solution of (15)
4: **if** $v_k = 0$ **then**
5: **return** x^k
6: **else**
7: **if** $k = 0$ **then**
8: Set $d_k = v_k$
9: **else**
10: Compute s_k as in (14)
11: Update $H_k(v_k)$ and $q_k(v_k)$ as in (23) and (33), respectively
12: Update

$$\tilde{s}_k := v_k - \frac{q_k(v_k) \left\| v_k^2 \right\|}{s_k^\top H_k(v_k)} s_k$$

13: Update $H_k(\tilde{s}_k)$ and $q_k(\tilde{s}_k)$ as in (23) and (33), respectively
14: Compute α_1^k and α_2^k as the minimizer of (24) and (25), respectively
15: Update $\tilde{d}_k := \alpha_1^k v_k + \alpha_2^k \tilde{s}_k$
16: Update $q_{k,d}$ as in (27)
17: Compute α_3^k as the minimizer of (26)
18: Update $d_k := \alpha_3^k \tilde{d}_k$
19: **end if**
20: Compute the stepsize $t_k \in (0, 1]$ in the following way:

$$t_k := \max \left\{ \gamma^j : j \in \mathbb{N},\ F\left(x^k + \gamma^j d_k\right) - F(x^k) \right. \\ \left. \leq \sigma \gamma^j (\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)). \right\}$$

21: Update $x^{k+1} := x^k + t_k d_k$
22: **end if**
23: **end for**

Remark 3 Lines 3, 10, 14, and 17 contribute to the main computational cost of Algorithm 1, since each of these steps requires solving a subproblem. Fortunately, due to the specific choice of P_k, the subproblems appearing in Lines 3 and 10 admit closed-form solutions. The remaining challenge lies in solving the three one-dimensional subproblems (24), (25), and (26). In the next subsection, we present efficient methods for solving these one-dimensional subproblems.

6.1 Methods for one-dimensional subproblem

The subproblems arising in Algorithm 1 reduce to several one-dimensional optimization problems. Depending on the structure of the objective function and the constraints, these problems can be categorized into two types: constrained quadratic optimization problems and ℓ_1-regularized optimization problems. In the following, we present specialized solution strategies for each case.

6.1.1 Constained optimization problems

The one-dimensional subproblem takes the form:

$$\min_{x^k+td_k\in\Omega} at^2+bt,$$

where $a>0$ and $b\in\mathbb{R}$ are constants. Note that there exist t_l and t_u such that

$$x^k+td_k\in\Omega,\ \forall t\in[t_l,t_u].$$

The optimal solution is therefore given by

$$t_k=\min\{\max\{t_l,-\frac{b}{2a}\},t_u\}.$$

We summarize the procedure in the following algorithm.

Algorithm 2: Method for one-dimensional constrained optimzation problem

Require: $x^k, d_k\in\mathbb{R}^n$, a, b, Ω
1: Compute $t_k=-\frac{b}{2a}$
2: **if** $x^k+t_kd_k\in\Omega$ **then**
3: **return** t_k
4: **else**
5: **return** $t_k=-\frac{b}{2a}\max\{t: x^k+t(-\frac{b}{2a}d_k)\in\Omega\}$
6: **end if**

6.1.2 ℓ_1 regularized problems

Another type of subproblem arising in Algorithm 1 involves ℓ_1 regularization and takes the form:

$$\min h(t):=at^2+bt+\|v+td\|_1,$$

where $a>0, b\in\mathbb{R}$ and $v,d\in\mathbb{R}^m$. The optimality condition for this problem is

$$0\in\partial h(t):=2at+b+\sum_{i\in[m]}\partial|v_i+td_i|,$$

where

$$\partial|v_i+td_i|:=\begin{cases}d_i, & v_i+td_i>0,\\ [-d_i,d_i], & v_i+td_i=0,\\ -d_i, & v_i+td_i<0.\end{cases}\tag{34}$$

Since h is convex, the subdifferential satisfies

$$\partial^+h(t_1)\le\partial^-h(t_2),\quad t_1<t_2.$$

Moreover, we have

$$\partial h(t)\subset[2at+b-D,2at+b+D],$$

where $D:=\sum_{i\in[m]}|d_i|$. Based on this property, we develop a partitioning method with the initial interval

$$\left[\frac{-D-b}{2a},\frac{D-b}{2a}\right].$$

Algorithm 3: Partitioning method for one-dimensional ℓ_1 regularized problem

Require: $L = \frac{-D-b}{2a}, U = \frac{D-b}{2a}, S = [L, U] \cap \{-\frac{v_i}{d_i} : d_i \neq 0, i \in [m]\} \cup \{L, U\}$, $Bool = \texttt{True}$

while $Bool$ **do**
  if $|S| \leq 2$ **then**
    $Bool = \texttt{False}$
    Compute $s = -(b + \sum_{i\in[m]} \texttt{sign}(v_i + \frac{L+U}{2} d_i) d_i)/(2a)$
    return $\max\{L, \min\{s, U\}\}$
  else
    Select s_{median} the median of S
    Compute the subdifferential $\partial h(s_{median})$
    if $0 \in \partial h(s_{median})$ **then**
      $Bool = \texttt{False}$
      return s_{median}
    else
      if $\partial^+ h(s_{median}) < 0$ **then**
        Update $L = s_{median}$
        Update $S = S_{[s \geq s_{median}]}$
      else
        Update $U = s_{median}$
        Update $S = S_{[s \leq s_{median}]}$
      end if
    end if
  end if
end while

6.2 Convergence analysis

In this section we analyze the convergence properties of the proposed algorithm. We first show that the stepsize produced by the line-search procedure admits a uniform lower bound.

Lemma 2 *Suppose that f is L-smooth. The stepsize generated by Algorithm 1 has a lower bound:*

$$t_{\min} := \min\{(1-\sigma)c_1/L, 1\}. \tag{35}$$

Proof It suffices to consider the case $t_k < 1$, in which the backtracking procedure is activated. In this situation the Armijo condition is violated for the trial stepsize $2t_k$, yielding

$$F\left(x^k + 2t_k d_k\right) - F(x^k) > 2\sigma t_k(\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)). \tag{36}$$

Since f is L-smooth, we have

$$\begin{aligned} &F\left(x^k + 2t_k d_k\right) - F(x^k) \\ &\leq 2t_k \nabla f(x^k)^\top d_k + g(Ax^k + 2t_k Ad_k) - g(Ax^k) + \frac{L}{2}\|2t_k d_k\|^2 \\ &\leq 2t_k(\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)) + \frac{L}{2}\|2t_k d_k\|^2, \end{aligned} \tag{37}$$

where the second inequality follows from the convexity of g and the fact that $2t_k \in (0,1]$. Combining this inequality with (36) gives

$$(\sigma-1)(\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)) \leq Lt_k \|d_k\|^2.$$

Using condition (31), we obtain

$$t_k \geq \frac{(1-\sigma)c_1}{L}. \tag{38}$$

Therefore $t_k \geq t_k^{\min}$, which completes the proof. □

To establish global convergence, we impose the following standard assumption on the objective function.

Assumption 1 *For any $x^0 \in \mathrm{dom} F$, the level set $\mathcal{L}_F(x^0) := \{x : F(x) \leq F(x^0)\}$ is compact.*

Under this assumption we can prove the global convergence of the proposed algorithm.

Theorem 1 *Suppose that Assumption 1 holds and f is L-smooth. Let $\{x^k\}$ be the sequences of points generated by Algorithm 1. Then $\{x^k\}$ has at least one accumulation point, and any accumulation point x^* is a stationary point.*

Proof By armijo line search, we deduce that $\{F(x^k)\}$ is monotone decreasing and

$$F(x^{k+1}) - F(x^k) \leq \sigma t_k(\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)) \leq -\frac{c_3 \min\{1, c_3/c_2\}}{4}\sigma t_k \|v_k\|^2, \tag{39}$$

where the last inequality follows by relation (32). Therefore $x^k \in \mathcal{L}_F(x^0)$ for all k, and hence $\{x^k\}$ has at least one accumulation point x^* due to the compactness of $\mathcal{L}_F(x^0)$. In particular, there exists an infinite index set $\mathcal{K}$ such that

$$\lim_{k \in \mathcal{K}} x^k = x^*.$$

Moreover, since F is lower semicontinuous and $\mathcal{L}_F(x^0)$ is compact, the sequence $\{F(x^k)\}$ is bounded below. Together with the monotonicity of $\{F(x^k)\}$, this implies that $\{F(x^k)\}$ is a Cauchy sequence. Hence

$$\lim_{k\to\infty} F(x^{k+1}) - F(x^k) = 0.$$

Combining this limit with (39) yields

$$\lim_{k\to\infty} t_k \|v_k\|^2 = 0. \tag{40}$$

Together with (38), we obtain

$$\lim_{k\to\infty} v_k = 0.$$

Recalling that $P_k \succeq c_3 \boldsymbol{I}_n$, we conclude that x^* is a stationary point. □

The above theorem guarantees that the proposed algorithm globally converges to stationary points. Next, we further strengthen the result by establishing a linear convergence rate under the stronger assumption that the objective function is strongly convex.

Theorem 2 *Suppose that f is L-smooth and strongly convex with module $\mu > 0$. Let $\{x^k\}$ be the sequence generated by Algorithm 1. Then*

$$F(x^{k+1}) - F(x^*) \leq \left(1 - \frac{\sigma}{2}\min\left\{\frac{\mu}{c_4}, 1\right\}\min\left\{\frac{c_3}{c_2}, 1\right\}\min\left\{\frac{(1-\sigma)c_1}{L}, 1\right\}\right)(F(x^k) - F(x^*)).$$

Proof By direct calculation, we have

$$
\begin{aligned}
F(x^k) - F(x^*) &\le \max_{x\in\mathbb{R}^n} \left\{ \nabla f(x^k)^\top (x^k - x) + g(Ax^k) - g(Ax) - \frac{\mu}{2} \left\| x - x^k \right\|^2 \right\} \\
&\le \max_{x\in\mathbb{R}^n} \left\{ \nabla f(x^k)^\top (x^k - x) + g(Ax^k) - g(Ax) - \frac{\mu}{2c_4} \left\| x - x^k \right\|_{P_k}^2 \right\} \\
&\le \max\left\{ \frac{c_4}{\mu}, 1 \right\} \max_{x\in\mathbb{R}^n} \left\{ \nabla f(x^k)^\top (x^k - x) + g(Ax^k) - g(Ax) - \frac{1}{2} \left\| x - x^k \right\|_{P_k}^2 \right\} \\
&\le \max\left\{ \frac{c_4}{\mu}, 1 \right\} \frac{1}{2} \left\| v_k \right\|_{P_k}^2 \\
&\le -2 \max\left\{ \frac{c_4}{\mu}, 1 \right\} \max\left\{ \frac{c_2}{c_3}, 1 \right\} (\nabla f(x^k)^\top d_k + g(Ax^k + Ad_k) - g(Ax^k)).
\end{aligned}
$$

Substituting this bound into line search condition gives

$$
F(x^{k+1}) - F(x^k) \le -\frac{1}{2} \min\left\{ \frac{\mu}{c_4}, 1 \right\} \min\left\{ \frac{c_3}{c_2}, 1 \right\} \sigma t_k (F(x^k) - F(x^*)).
$$

The desired result follows by rearranging the inequality and adding $-F(x^*)$ to both sides.

7 Numerical experiments

In this section, we evaluate the empirical performance of the proposed method on several composite optimization problems. The goal of these experiments is to assess the efficiency and robustness of the proposed algorithm in comparison with several widely used first-order methods.

The tested algorithms are summarized as follows:

- FISTA_bt: FISTA with backtracking [23];
- FISTA_bt_rs: FISTA with backtracking and gradient restart [18];
- PDHG: primal-dual hybird gradient method [9]:

$$
\begin{aligned}
x^{k+1} &= x^k - \tau \nabla f(y^k), \\
\bar{x}^k &= x^{k+1} + \theta(x^{k+1} - x^k), \\
y^{k+1} &= \Pi_{\mathrm{B}_\infty[\mathbf{0}_m,\lambda]}(y^k + \sigma A\bar{x}),
\end{aligned}
$$

  where $\tau = 1/L$, $\theta = 0.9$ and $\sigma = 4/(\tau(1+\theta)^2\|A^\top A\|)$;
- P^2GM_M: the preconditioned proximal gradient method with momentum, obtained from Algorithm 1 by removing Line 12, i.e., $\tilde{s}_k = s_k$;
- P^2GM_CM: preconditioned proximal gradient method with subspace acceleration, as described in Algorithm 1.

In P^2GM_M and P^2GM_CM, the preconditioning matrix is chosen as $P_k = \alpha_k P$, where P is defined as in (6). The scalar α_k is selected according to a Barzilai–Borwein stepsize computed in the P-norm, which allows the scaling of the preconditioner to adapt to the local curvature of the objective function.

All numerical experiments were implemented in Python 3.7 and conducted on a personal computer equipped with an Intel Core i7-11390H processor (3.40 GHz) and 16 GB of RAM. To evaluate the convergence behavior of the tested algorithms, we compute an approximation of the optimal objective value by running P^2GM_CM for a sufficiently large number of iterations. Denote this value by $\tilde{F} \approx F^*$. For each algorithm we then report the objective gap $F(x^k) - \tilde{F}$.

In the following subsections, we present numerical results on three representative applications: LASSO problems, simplex-constrained quadratic problems, and structured quadratic composite problems.

7.1 LASSO problems

We first consider the LASSO problem, which is a widely used benchmark in sparse signal recovery and machine learning. The problem is given by

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1. \tag{41}$$

Here $A\in\mathbb{R}^{m\times n}$ is the data matrix, $b\in\mathbb{R}^m$ is the observation vector, and $\lambda>0$ is a regularization parameter controlling the sparsity of the solution.

The problem instance is synthetically generated to simulate a high-dimensional, noisy, and numerically ill-conditioned sparse regression environment.

- **Dimensions:** We set the number of samples $m=5000$ and the feature dimension $n=500$.
- **Ill-Conditioned Design Matrix:** The matrix A is constructed using the singular value decomposition

  $$A=U\Sigma V^\top.$$

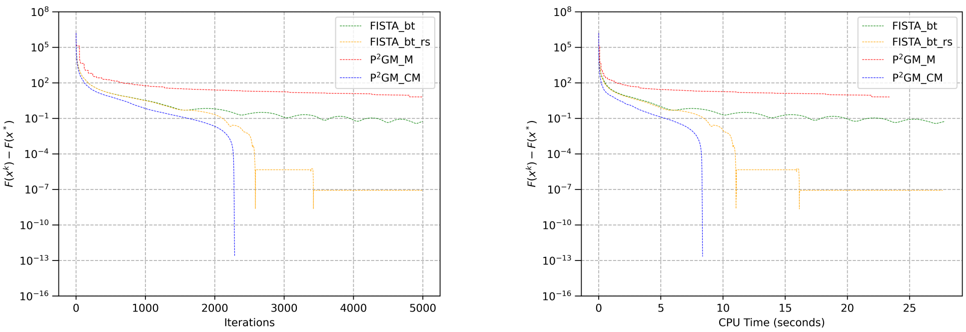
  To induce numerical stiffness, the singular values are logarithmically spaced in the interval $[10^{-3},1]$ and scaled by $\sqrt{m}$. This scaling ensures that the data term

  $$\frac{1}{m}A^\top A$$

  has a bounded spectral norm while maintaining a large condition number $\kappa\approx 10^6$. The orthogonal matrices U and V are generated from random Gaussian matrices via QR factorization.
- **Sparse Ground Truth:** The true solution x_{true} is generated with extreme sparsity ($p=0.5\%$), resulting in approximately 2–3 active nonzero elements drawn uniformly from $[1,2]$. This sparsity forces the optimization trajectory to frequently interact with the non-smooth boundaries induced by the ℓ_1 regularization.
- **Noisy Observations:** The observation vector is generated as

  $$b=Ax_{\text{true}}+\varepsilon,$$

  where ε is Gaussian noise with noise level 10^{-3}. This introduces measurement noise and simulates realistic perturbations in the data.



(a) Iterations (b) CPU time

Fig. 1: Objective gaps w.r.t. iterations and CPU time for problem (41) with $\lambda = 10^{-4}$.

7.2 Ill-conditioned quadratic problems with simplex constraint

Next, we consider the minimization of a quadratic function over the unit simplex:

$$\min_{x \in \Delta_n} f(x) = \frac{1}{2} x^\top Q x + c^\top x. \tag{42}$$

This problem serves as a representative example of constrained optimization with a highly ill-conditioned objective function.

The problem instance is generated according to the following specifications:

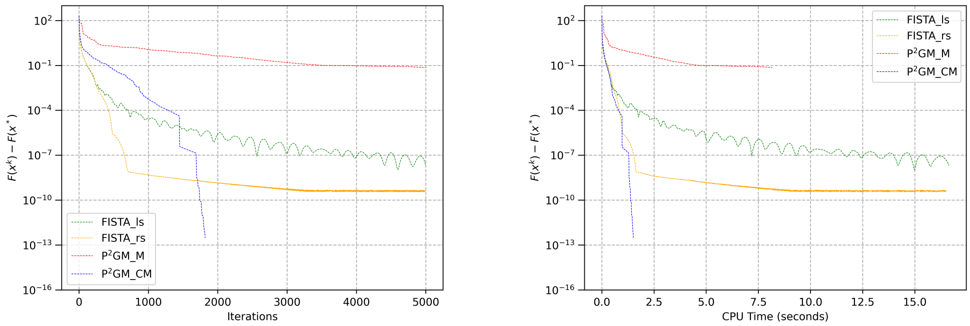
- **Dimension:** $n = 100$.
- **Condition Number:**

$$\kappa = 5 \times 10^5.$$

- **Spectrum:** The eigenvalues of Q are logarithmically spaced in the interval $[1, \kappa]$.
- **Matrix Construction:** The positive definite matrix Q is constructed via spectral decomposition

$$Q = U \Lambda U^\top,$$

  where U is a random orthogonal matrix generated from a Gaussian matrix via QR factorization and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the prescribed eigenvalues.
- **Linear Term:** The vector c is sampled from a standard Gaussian distribution.



(a) Iterations (b) CPU time

Fig. 2: Objective gaps w.r.t. iterations and CPU time for problem (42).

7.3 Structured ℓ_1 regularization quadratic problems

Finally, we consider a structured composite optimization problem of the form

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + \lambda\|Ax\|_1, \tag{43}$$

where the smooth component is the quadratic function

$$f(x) = \frac{1}{2}x^\top Qx + c^\top x.$$

This problem combines an ill-conditioned quadratic objective with a structured linear operator in the nonsmooth term.

The problem parameters are generated as follows:

- **Dimension:** $n = 100$, and the linear operator A has $m = 50$ rows.
- **Quadratic Term:** The matrix $Q \in \mathbb{S}^n_{++}$ is generated via spectral decomposition

$$Q = U\Lambda U^\top,$$

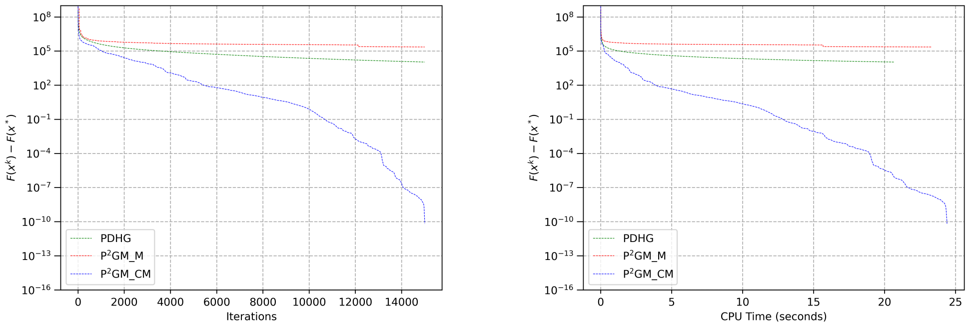
  where U is a random orthogonal matrix obtained from the QR factorization of a Gaussian matrix. The eigenvalues $\{\lambda_i\}$ are logarithmically spaced in the interval $[1,\kappa]$ with $\kappa = 5\times 10^4$.
- **Linear Term:** The vector $c \in \mathbb{R}^n$ is sampled from a standard Gaussian distribution.
- **Linear Operator:** The matrix $A \in \mathbb{R}^{m\times n}$ is constructed via singular value decomposition

$$A = U_A\Sigma V^\top,$$

  where U_A and V are random orthogonal matrices generated from Gaussian matrices via QR factorization. The singular values of A are logarithmically spaced in $[1,\sigma_A]$ with $\sigma_A = \sqrt{5000}$.
- **Preconditioning Matrix:** To exploit the structure of A, we construct the preconditioning matrix

$$P = V\left(\Lambda^\top\Lambda + \begin{bmatrix}\mathbf{0}_{m\times m} & \\ & \tilde{P}\end{bmatrix}\right)V^\top,$$

  where $\tilde{P} \in \mathbb{S}^{n-m}_{++}$ is chosen as the identity matrix. The inverse P^{-1} can therefore be computed analytically from the block structure.



(a) Iterations (b) CPU time

Fig. 3: Objective gaps w.r.t. iterations and CPU time for problem (43) with $\lambda = 1/16$.

7.4 Discussion of numerical results

Figures 1–3 illustrate the convergence behavior of the tested algorithms on three representative problems: LASSO, simplex-constrained quadratic programming, and structured ℓ_1 regularization problems. In all experiments, the proposed method $\text{P}^2\text{GM\_CM}$ consistently achieves the fastest decrease of the objective gap with respect to both iterations and CPU time.

A key observation from these figures is the clear performance gap between $\text{P}^2\text{GM\_CM}$ and its non-conjugate variant $\text{PP}^2\text{GM\_M}$. While both methods share the same preconditioning framework, the orthogonalized conjugate momentum used in $\text{P}^2\text{GM\_CM}$ significantly improves the efficiency of the subspace search, leading to much faster convergence. This confirms that incorporating curvature information through conjugate directions plays an essential role in accelerating the algorithm.

Moreover, compared with classical first-order methods such as FISTA and PDHG, the proposed approach demonstrates stronger robustness to ill-conditioning and maintains stable convergence across different problem structures. These results highlight the advantage of combining structure-exploiting preconditioning with subspace acceleration.

8 Conclusions

In this paper we studied composite optimization problems involving linear operators and proposed a preconditioned proximal gradient framework with conjugate momentum from a subspace perspective. By exploiting the structure of the linear operator, we constructed a preconditioner that transforms the proximal gradient subproblem into a dual formulation whose solution admits closed-form expressions for several important classes of problems. This structure-driven preconditioning significantly simplifies the computation of proximal updates.

To further improve the convergence behavior, we introduced a subspace acceleration mechanism that incorporates curvature information through a proximal Newton model restricted to a low-dimensional subspace. By orthogonalizing the subspace basis with respect to the Hessian-induced inner product, the resulting two-dimensional nonsmooth problem can be efficiently approximated by a sequence of one-dimensional optimization problems. This design preserves computational efficiency while capturing useful second-order information.

We established global convergence of the proposed algorithm and proved a Q-linear convergence rate under standard smoothness and strong convexity assumptions. Numerical experiments on several representative applications, including LASSO problems, simplex-constrained quadratic programs, and structured ℓ_1 regularization problems, demonstrated that the proposed method achieves competitive performance, especially on ill-conditioned problems.

Several directions for future research remain open.

- First, it would be of interest to extend the proposed framework to more general composite optimization problems in which the nonsmooth term may also be nonconvex. Such an extension would broaden the applicability of the method to a wider class of modern machine learning and signal processing models.
- Second, inspired by the recent developments of the dimension-reduced second-order method [28], it would be worthwhile to investigate the fast asymptotic convergence properties of the proposed algorithm. In particular, understanding whether the subspace proximal framework can inherit superlinear or fast local convergence behavior remains an interesting theoretical question.
- Finally, our numerical experiments indicate that the orthogonalization of the momentum direction plays a crucial role in improving the practical performance of the algorithm. This observation suggests a broader research direction: incorporating conjugate or orthogonalized momentum into other momentum-based optimization frameworks. In particular, it would be interesting to explore whether similar ideas can be integrated into accelerated schemes such as Nesterov's method, potentially leading to new accelerated algorithms with improved robustness and convergence behavior.

References

1. H. Attouch, Z. Chbani, J. Fadili, H. Riahi. First-order optimization algorithms via inertial systems with Hessian driven damping. Math. Program., 193:113–155, 2022.
2. S. Becker, M. J. Fadili. A quasi-Newton proximal splitting method. In Advances in Neural Information Processing Systems, pages 2627–2635, 2012.
3. A. Beck, M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci., 2(1):183–202, 2009.
4. A. Chambolle, T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vision, 40:120–145, 2011.
5. J. Chen, L. Tang, X. M. Yang. A subspace minimization Barzilai-Borwein method for multiobjective optimization problems. Comput. Optim. Appl., 92:155–178, 2025.
6. L. Condat. Fast projection onto the simplex and the ℓ_1 ball. Math. Program., 158:575–585, 2016.
7. Y. Dai, Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim., 10(1):177–182, 1999.
8. R. Fletcher, C. Reeves. Function minimization by conjugate gradients. Comput. J., 7(2):149–154, 1964.
9. B. He, X. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci., 5(1):119–149, 2012.
10. M. R. Hestenes, E. Stiefel. Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand., 49(6):409–436, 1952.
11. M. Lapucci, G. Liuzzi, S. Lucidi, M. Sciandrone. A globally convergent gradient method with momentum. Comput. Optim. Appl., 93:795–820, 2026.
12. M. Lapucci, G. Liuzzi, S. Lucidi, M. Sciandrone, D. Scuppa. Projected gradient methods with momentum. arXiv preprint arXiv:2601.16683, 2026.

13. J. Lee, Y. Sun, M. Saunders. Proximal Newton-type methods for minimizing composite functions. SIAM J. Optim., 24(3):1420–1443, 2014.
14. H. Luo, L. Chen. From differential equation solvers to accelerated first-order methods for convex optimization. Math. Program., 195:735–781, 2022.
15. Y. Malitsky, T. Pock. A first-order primal-dual algorithm with linesearch. SIAM J. Optim., 28(1):411–432, 2018.
16. Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Soviet Math. Dokl., 27(2):372–376, 1983.
17. Y. Nesterov. Gradient methods for minimizing composite objective function. Math. Program., 140:125–161, 2013.
18. B. O'Donoghue, E. Candès. Adaptive restart for accelerated gradient schemes. Found. Comput. Math., 15:715–732, 2015.
19. Y. Park, S. Dhar, S. Boyd, M. Shah. Variable metric proximal gradient method with diagonal Barzilai-Borwein stepsize. In ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, 3597–3601, 2020.
20. B. T. Polyak. Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys., 4(5):1–17, 1964.
21. E. Polak, G. Ribière. Note sur la convergence de méthodes de directions conjuguées. Rev. Française Inform. Rech. Opér., 3:35–43, 1969.
22. Z. Qu, W. Gao, O. Hinder, Y. Ye, Z. Zhou. Optimal diagonal preconditioning. Oper. Res., 73(3):1479–1495, 2025.
23. K. Scheinberg, D. Goldfarb, X. Bai. Fast first-order methods for composite convex optimization with backtracking. Found. Comput. Math., 14:389–417, 2014.
24. K. Scheinberg, X. Tang. Practical inexact proximal quasi-Newton method with global complexity analysis. Math. Program., 160(1):495–529, 2016.
25. B. Shi, S. Du, W. Su, M. Jordan. Understanding the acceleration phenomenon via high-resolution differential equations. Math. Program., 195:79–148, 2022.
26. W. Su, S. Boyd, E. Candès. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. J. Mach. Learn. Res., 17(153):1–43, 2016.
27. Y. Yuan, J. Stoer. A subspace study on conjugate gradient algorithms. Z. Angew. Math. Mech., 75(1):69–77, 1995.
28. C. Zhang, D. Ge, C. He, B. Jiang, Y. Jiang, Y. Ye. DRSOM: A dimension reduced second-order method. arXiv preprint arXiv:2208.00208, 2022.

Acknowledgements This work was funded by the National Key Research and Development Program of China [grant number 2023YFA1011504]; the Major Program of the National Natural Science Foundation of China [grant numbers 11991020, 11991024]; the Key Program of the National Natural Science Foundation of China [grant number 12431010]; the General Program of the National Natural Science Foundation of China [grant number 12171060]; NSFC-RGC (Hong Kong) Joint Research Program [grant number 12261160365]; the Team Project of Innovation Leading Talent in Chongqing [grant number CQYC20210309536]; the Natural Science Foundation of Chongqing [grant numbers ncamc2022-msxm01, CSTB2024NSCQ-LZX0140]; Major Project of Science and Technology Research Rrogram of Chongqing Education Commission of China [grant number KJZD-M202300504]; the Chongqing Postdoctoral Research Project Special Grant [grant number 2024CQBSHTB1007]; the Science and Technology Research Program of Chongqing Education commission of China [grant number KJQN202400520] and Foundation of Chongqing Normal University [grant numbers 22XLB005, 22XLB006].