

ON QUASI-NEWTON FORWARD-BACKWARD SPLITTING: PROXIMAL CALCULUS AND CONVERGENCE

STEPHEN BECKER*, JALAL FADILI†, AND PETER OCHS‡

Abstract. We introduce a framework for quasi-Newton forward-backward splitting algorithms (proximal quasi-Newton methods) with a metric induced by diagonal \pm rank- r symmetric positive definite matrices. This special type of metric allows for a highly efficient evaluation of the proximal mapping. The key to this efficiency is a general proximal calculus in the new metric. By using duality, formulas are derived that relate the proximal mapping in a rank- r modified metric to the original metric. We also describe efficient implementations of the proximity calculation for a large class of functions; the implementations exploit the piece-wise linear nature of the dual problem. Then, we apply these results to acceleration of composite convex minimization problems, which leads to elegant quasi-Newton methods for which we prove convergence. The algorithm is tested on several numerical examples and compared to a comprehensive list of alternatives in the literature. Our quasi-Newton splitting algorithm with the prescribed metric compares favorably against state-of-the-art. The algorithm has extensive applications including signal processing, sparse recovery, machine learning and classification to name a few.

Key words. forward-backward splitting, quasi-Newton, proximal calculus, duality.

AMS subject classifications. 65K05, 65K10, 90C25, 90C31.

1. Introduction. Convex optimization has proved to be extremely useful to all quantitative disciplines of science. A common trend in modern science is the increase in size of datasets, which drives the need for more efficient optimization schemes. For large-scale unconstrained smooth convex problems, two classes of methods have seen the most success: limited memory quasi-Newton methods and non-linear conjugate gradient (CG) methods. Both of these methods generally outperform simpler methods, such as gradient descent. However, many problems in applications have constraints or should be modeled naturally as non-smooth optimization problems.

A problem structure that is sufficiently broad to cover many applications in machine learning, signal processing, image processing, computer vision (and many others) is the minimization of the sum of two convex function, one being smooth and the other being non-smooth and “simple” in a certain way. The gradient descent method has a natural extension to these structured non-smooth optimization problems, which is known as *proximal gradient descent* (which includes projected gradient descent as a sub-case) or *forward-backward splitting* [1]. Algorithmically, besides a gradient step with respect to the smooth term of the objective, the generalization requires to solve proximal subproblems with respect to the non-smooth term of the objective. The property “simple” from above refers the proximal subproblems. In many situations, these subproblems can be solved analytically or very efficiently. However, a change of the metric, which is the key feature of quasi-Newton methods or non-linear CG, often leads to computationally hard subproblems.

While the convergence of proximal quasi-Newton methods has been analyzed to some extent in the context of variable metric proximal gradient methods, little attention is paid to the efficient evaluation of the subproblems in the new metric. In this paper, we emphasize the fact that quasi-Newton methods construct a metric with a special structure: the metric is successively updated using low rank matrices. We develop efficient calculus rules for a general rank- r modified metric. This allows popular quasi-Newton methods, such as the SR1

*Applied Mathematics, University of Colorado Boulder (stephen.becker@colorado.edu).

†Normandie Univ, ENSICAEN, CNRS, GREYC, France (Jalal.Fadili@greyc.ensicaen.fr).

‡Saarland University, Saarbrücken, Germany (ochs@math.uni-sb.de).

(symmetric rank-1) and the L-BFGS methods, to be efficiently applied to structured non-smooth problems. The SR1 method pursues a rank-1 update of the metric and the L-BFGS method uses a rank-2 update.

We consider the results in this paper as a large step toward the applicability of quasi-Newton methods with a comparable efficiency for smooth and structured non-smooth optimization problems.

1.1. Problem statement. Let $\mathcal{H} = (\mathbb{R}^N, \langle \cdot, \cdot \rangle)$ equipped with the usual Euclidean scalar product $\langle x, y \rangle = \sum_{i=1}^N x_i y_i$ and associated norm $\|x\| = \sqrt{\langle x, x \rangle}$. For a matrix $V \in \mathbb{R}^{N \times N}$ in the symmetric positive-definite (SDP) cone $\mathbb{S}_{++}(N)$, we define $\mathcal{H}_V = (\mathbb{R}^N, \langle \cdot, \cdot \rangle_V)$ with the scalar product $\langle x, y \rangle_V = \langle x, Vy \rangle$ and norm $\|x\|_V$ corresponding to the metric induced by V . The dual space of \mathcal{H}_V , under $\langle \cdot, \cdot \rangle$, is $\mathcal{H}_{V^{-1}}$. We denote Id the identity operator.

An extended-valued function $f : \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ is (0)-*coercive* if $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$. The *domain* of f is defined by $\text{dom } f = \{x \in \mathcal{H} : f(x) < +\infty\}$ and f is *proper* if $\text{dom } f \neq \emptyset$. We say that a real-valued function f is *lower semi-continuous* (lsc) if $\liminf_{x \rightarrow x_0} f(x) \geq f(x_0)$. The class of all proper lsc convex functions from \mathcal{H} to $\mathbb{R} \cup \{+\infty\}$ is denoted by $\Gamma_0(\mathcal{H})$. The conjugate or Legendre-Fenchel transform of f on \mathcal{H} is denoted f^* .

Our goal is the generic minimization of functions of the form

$$\min_{x \in \mathcal{H}} \{F(x) := f(x) + h(x)\}, \quad (\text{P})$$

where $f, h \in \Gamma_0(\mathcal{H})$. We also assume the set of minimizers $\text{Argmin}(F)$ is nonempty. Write x^* to denote an element of $\text{Argmin}(F)$. We assume that $f \in C^{1,1}(\mathbb{R}^N)$, meaning that it is continuously differentiable and its gradient (in \mathcal{H}) is L -Lipschitz continuous.

The class we consider covers structured smooth+non-smooth convex optimization problems, including those with convex constraints. Here are some examples in regression, machine learning and classification.

EXAMPLE 1.1 (LASSO). *Let A be a matrix, b a vector, and $\lambda > 0$ of appropriate dimensions.*

$$\min_{x \in \mathcal{H}} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1. \quad (1.1)$$

EXAMPLE 1.2 (Non-negative least-squares (NNLS)). *Let A and b be as in Example 1.1.*

$$\min_{x \in \mathcal{H}} \frac{1}{2} \|Ax - b\|_2^2 \quad \text{subject to } x \geq 0. \quad (1.2)$$

EXAMPLE 1.3 (Sparse Support Vector Machines). *One would like to find a linear decision function which minimizes the objective*

$$\min_{x \in \mathcal{H}, b \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\langle x, z_i \rangle + b, y_i) + \lambda \|x\|_1 \quad (1.3)$$

where for $i = 1, \dots, m$, $(z_i, y_i) \in \mathbb{R}^N \times \{\pm 1\}$ is the training set, and \mathcal{L} is a smooth loss function with Lipschitz-continuous gradient such as the squared hinge loss $\mathcal{L}(\hat{y}_i, y_i) = \max(0, 1 - \hat{y}_i y_i)^2$ or the logistic loss $\mathcal{L}(\hat{y}_i, y_i) = \log(1 + e^{-\hat{y}_i y_i})$. The term $\lambda \|x\|_1$ promotes sparsity of the decisive features steered by a parameter $\lambda > 0$.

1.2. Contributions. We introduce an general proximal calculus in a metric $V = P \pm Q \in \mathbb{S}_{++}(N)$ given by $P \in \mathbb{S}_{++}(N)$ and a positive semi-definite rank- r matrix Q . This significantly extends the result in the preliminary version of this paper [2], where only $V = P + Q$ with a rank-1 matrix Q is addressed. The general calculus is accompanied by several more concrete examples (see Section 3.3.3 for a non-exhaustive list), where, for example, the piecewise linear nature of certain dual problems is rigorously exploited.

Motivated by the discrepancy between constrained and unconstrained performance, we define a class of limited-memory quasi-Newton methods to solve (P) and that extends naturally and elegantly from the unconstrained to the constrained case. In particular, we generalize the zero-memory SR1 and L-BFGS quasi-Newton methods to the proximal quasi-Newton setting for solving (P), and prove their convergence. Where L-BFGS-B [3] is only applicable to box constraints, our quasi-Newton methods efficiently apply to a wide-variety of non-smooth functions.

1.3. Paper organization. Section 2 formally introduces quasi-Newton methods and their generalization to the structured non-smooth setting (P). The related literature is extensively discussed. In order to obtain a clear perspective on how to apply the proximal calculus that is developed in Section 3, the outline of our proposed zero-memory SR1 and our zero-memory BFGS quasi-Newton method is provided in Section 2. The main result that simplifies the rank- r modified proximal mapping is stated in Section 3.2, followed by several specializations and an efficient semi-smooth root finding strategy that is required in some situations. Section 4 describes the details for the construction of the SR1 metric and states the convergence result. Following the same outline, the L-BFGS metric is constructed in Section 5 and convergence is proved. The significance of our results is confirmed in numerical experiments.

2. Quasi-Newton forward–backward splitting.

2.1. The algorithm. The main update step of our proposed algorithm for solving (P) is a forward–backward splitting (FBS) step in a special type of metric. In this section, we introduce the main algorithmic step and Section 3 shows that our choice of metric allows the update to be computed efficiently.

We define the following quadratic approximation to the smooth part f of the objective function in (P) around the current iterate x_k

$$Q_{\kappa}^B(x; x_k) := f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\kappa} \|x - x_k\|_B^2, \quad (2.1)$$

where $B \in \mathbb{S}_{++}(N)$ and $\kappa > 0$. The (non-relaxed) version of the variable metric FBS algorithm (also known as proximal gradient descent) to solve (P) updates to a new iterate x_{k+1} according to

$$x_{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^N} Q_{\kappa_k}^{B_k}(x; x_k) + h(x) =: \operatorname{prox}_{\kappa_k h}^{B_k}(x_k - \kappa_k B_k^{-1} \nabla f(x_k)) \quad (2.2)$$

with (iteration dependent) step size κ_k and metric $B_k \in \mathbb{S}_{++}(N)$. The right hand side uses the so-called proximal mapping, which is formally introduced in Definition 3.1. Standard results (see, e.g., [4, 5]) show that, for a sequence $(B_k)_{k \in \mathbb{N}}$ that varies moderately (in the Loewner partial ordering sense) such that $\inf_{k \in \mathbb{N}} \|B_k\| = 1$, convergence of the scheme iterates $(x_k)_{k \in \mathbb{N}}$ is expected when $0 < \underline{\kappa} \leq \kappa_k \leq \bar{\kappa} < 2/L$.

Note that when $h = 0$, (2.2) reduces to gradient descent if $B_k = \operatorname{Id}$, which is a poor approximation and requires many iterations, but each step is cheap. When f is also $C^2(\mathbb{R}^N)$,

the Newton’s choice $B_k = \nabla^2 f(x_k)$ is a more accurate approximation and reduces to Newton’s method when $h = 0$. The update step is well-defined if f is strictly convex, but may be computationally demanding as it requires solving a linear system and possibly storing the Hessian matrix. Yet, because it is a more accurate approximation, Newton’s method has local quadratic convergence under standard assumptions such as self-concordancy. Motivated by the superiority of Newton and quasi-Newton methods over gradient descent for the case $h = 0$, we pursue a quasi-Newton approximation for B_k . However, the update is now much more involved than just solving a linear system. Indeed, one has to compute the proximal mapping in the metric B_k , which is, in general, as difficult as solving the original problem (P). For this reason, we restricted B_k to the form of a diagonal matrix plus or minus a low-rank term.

The main steps of our general quasi-Newton forward–backward scheme to solve (P) are given in Algorithm 1. Its instantiation for a diagonal – rank 1 metric (OSR1) and a diagonal – rank 2 metric (OBFGS) are respectively listed in Algorithm 2 and Algorithm 3. Details for the selection of the corresponding metrics are provided in Section 4 and 5. Following the convention in the literature on quasi-Newton methods, throughout the paper, we use B_k as an approximation to the Hessian and $H_k := B_k^{-1}$ as inverse approximation. The algorithms are listed as simply as possible to emphasize the important components; the actual software used for numerical tests is open-source and available at <https://github.com/stephenbeckr/zeroSR1>.

Algorithm 1 Quasi-Newton forward–backward framework to solve (P)

Require: $x_0 \in \text{dom}(f + h)$, stopping criterion ϵ , method to compute stepsizes t and κ_k (e.g. based on the Lipschitz constant estimate L of ∇f and strong convexity μ of f)

- 1: **for** $k = 1, 2, 3, \dots$ **do**
- 2: $s_k \leftarrow x_k - x_{k-1}$
- 3: $y_k \leftarrow \nabla f(x_k) - \nabla f(x_{k-1})$
- 4: Compute H_k according to a quasi-Newton framework
- 5: Define $B_k = H_k^{-1}$ and compute the variable metric proximity operator (see Section 3) with stepsize κ_k

$$\bar{x}_{k+1} \leftarrow \text{prox}_{\kappa_k h}^{B_k}(x_k - \kappa_k H_k \nabla f(x_k)) \quad (2.3)$$

- 6: $p_k \leftarrow \bar{x}_{k+1} - x_k$ and terminate if $\|p_k\| < \epsilon$
 - 7: Line-search along the ray $x_k + tp_k$ to determine x_{k+1} , or choose $t = 1$.
 - 8: **end for**
-

Algorithm 2 Zero-memory Symmetric Rank 1 (OSR1) algorithm to solve (P), cf. Section 4

Require: as for Algorithm 1, and parameters $\gamma, \tau_{\min}, \tau_{\max}$ for Algorithm 5

Iterate as in Algorithm 1, with line 4 as:

- 4: Compute H_k via Algorithm 5 (diagonal plus rank one)
-

Algorithm 3 Zero-memory BFGS (0BFGS) algorithm to solve (P), cf. Section 5

Require: as for Algorithm 1

Iterate as in Algorithm 1, with line 4 as:

4: Compute H_k via Eq. (5.1) (diagonal plus rank two)

REMARK 2.1. *The usage of the terms “diagonal – rank r ” and “diagonal + rank r ” need clarification. The meaning of these terms is that $B_k = D - \sum_{i=1}^r u_i u_i^\top$ or $B_k = D + \sum_{i=1}^r u_i u_i^\top$, respectively, where D is a diagonal matrix and $u_i \in \mathbb{R}^N$. Collectively, to cover both cases, $B_k = D \pm \sum_{i=1}^r u_i u_i^\top$ is used. Algorithmically, the choice of “+” or “–” is crucial.*

For instance, if we talk about a “diagonal \pm rank 1 quasi-Newton method”, this taxonomy applies to the approximation of the Hessian B_k . Since, the inverse H_k can be computed conveniently with the Sherman–Morrison inversion lemma, it is also of type “diagonal \mp rank 1”, where the sign of the rank 1 part is flipped. The analysis in [2] of the rank 1 proximity operator applied to the case “diagonal + rank 1”. In this paper, we cover both cases “diagonal \pm rank 1”, which generalizes and formalizes the “diagonal – rank 1” setting in [6].

2.2. Relation to prior work.

First-order methods. The algorithm in (2.2) with $B_k = \text{Id}$ is variously known as proximal descent or iterated shrinkage/thresholding algorithm (IST or ISTA). It has a grounded convergence theory, and also admits over-relaxation factors $\alpha \in (0, 1)$ [7].

The spectral projected gradient (SPG) [8] method was designed as an extension of the Barzilai–Borwein spectral step-length method to constrained problems. In [9], it was extended to non-smooth problems by allowing general proximity operators; The Barzilai–Borwein method [10] uses a specific choice of step-length κ_k motivated by quasi-Newton methods. Numerical evidence suggests the SPG/SpaRSA method is highly effective, although convergence results are not as strong as for ISTA.

FISTA [11] is a (two-step) inertial version of ISTA inspired by the work of Nesterov [12]. It can be seen as an explicit-implicit discretization of a nonlinear second-order dynamical system (oscillator) with viscous damping that vanishes asymptotically in a moderate way [13, 14]. While the stepsize κ is chosen in a similar way to ISTA (though with a smaller upper-bound), in our implementation, we tweak the original approach by using a Barzilai–Borwein step size, a standard line search, and restart [15], since this led to improved performance.

Recently, [16] has shown that optimizing the inertial parameter in each iteration of FISTA, applied to the sum of a quadratic function and a non-smooth function, the method is equivalent to the zero memory SR1 proximal quasi-Newton method that we propose in Section 4. Convergence is analyzed with respect to standard step sizes that relate to the Lipschitz constant, which does not cover the case of Barzilai–Borwein step size.

The above approaches assume B_k is a constant diagonal. The general diagonal case was considered in several papers in the 1980s as a simple quasi-Newton method, but never widely adapted. Variable metric operator splitting methods have been designed to solve monotone inclusion problems and convex minimization problems, see for instance [4, 5] in the maximal monotone case and [17] for the strongly monotone case. The convergence proofs rely on a variable metric extension of quasi-Fejér monotonicity [18]. In particular, this requires the variable metric to be designed a priori to verify appropriate growth conditions. However, it is not clear how to make the metric adapt to the geometry of the problem. In fact, in practice, the metric is usually chosen to be diagonal for the proximity operator to be easily computable. When the metric is not diagonal but fixed, these methods can be viewed as

pre-conditioned versions that were shown to perform well in practice for certain problems (i.e. functions h) [19, 20]. But again, the choice of the metric (pre-conditioner) is quite limited for computational and storage reasons.

Active set approaches. Active set methods take a simple step, such as gradient projection, to identify active variables, and then uses a more advanced quadratic model to solve for the free variables. A well-known such method is L-BFGS-B [3, 21] which handles general box-constrained problems; we test an updated version [22]. A recent bound-constrained solver is ASA [23] which uses a conjugate gradient (CG) solver on the free variables, and shows good results compared to L-BFGS-B, SPG, GENCAN and TRON. We also compare to several active set approaches specialized for ℓ_1 penalties: “Orthant-wise Learning” (OWL) [24], “Projected Scaled Sub-gradient + Active Set” (PSSas) [25], “Fixed-point continuation + Active Set” (FPC_AS) [26], and “CG + IST” (CGIST) [27].

Other approaches. By transforming the problem into a standard conic programming problem, the generic problem is amenable to interior-point methods (IPM). IPM requires solving a Newton-step equation, so first-order like “Hessian-free” variants of IPM solve the Newton-step approximately, either by approximately solving the equation or by subsampling the Hessian. The main issues are speed and robust stopping criteria for the approximations.

Yet another approach is to include the non-smooth h term in the quadratic approximation. Yu et al. [28] propose a non-smooth modification of BFGS and L-BFGS, and test on problems where h is typically a hinge-loss or related function. Although convergence of this method cannot be expected in general, there are special cases for which convergence results could be established [29, 30], and more recently [31]. The empirically justified good numerical performance has been observed for decades [32].

The projected quasi-Newton (PQN) algorithm [33, 34] is perhaps the most elegant and logical extension of quasi-Newton methods, but it involves solving a sub-iteration or need to be restricted to a diagonal metric in the implementation [35, 36]. PQN proposes the SPG [8] algorithm for the subproblems, and finds that this is an efficient trade-off whenever the cost function (which is not involved in the sub-iteration) is significantly more expensive to evaluate than projecting onto the constraints. Again, the cost of the sub-problem solver (and a suitable stopping criteria for this inner solve) are issues. The paper [37] shows how the sub-problem can be solved efficiently by a special interior-point method when h is a quadratic-support function. As discussed in [38], it is possible to generalize PQN to general non-smooth problems whenever the proximity operator is known (since, as mentioned above, it is possible to extend SPG to this case). In the same line of methods, [39] proposes a flexible proximal quasi-Newton method that extends [36] to simple proximal operators, though a diagonal metric is considered in the implementation. Another work that unifies and generalizes several of the works mentioned above in a variable metric (i.e. quasi-Newton) setting is [40].

A more general and efficient step size strategy with memory was proposed in [41] for unconstrained optimization, which was generalized to a scaled gradient projection method in [42], and used in the proximal gradient method in [43]. However, the flexible choice of the step size and the scaling of the metric is not for free when convergence guarantees are sought. [43, 39] rely on a line search strategy to account for a descent of the objective values. The metric in [44] is constructed such that (2.1) is a majorizer of the (possibly non-convex) objective and the step size selection is more conservative, however line search can be avoided.

The works [45, 46] make use of the so-called forward–backward envelope, a concept that allows them to reinterpret the forward–backward splitting algorithm as a variable metric gradient method for a smooth optimization problem. Using this reformulation, they can apply classical Newton or quasi-Newton methods. Proximal quasi-Newton methods have also

been considered in combination with the Heavy-ball method [47], and have been generalized further.

The proximal quasi-Newton methods described so far simply assume that the scaled proximal mapping can be solved efficiently, rely on solving subproblems, or simple diagonal scaling metrics. The first work on systematically solving non-diagonally scaled proximal mappings efficiently is the conference version of this paper [2]. The key is structure of the metric. In [2], it is assumed to be given as the sum of a diagonal and a rank-1 matrix. For the special case of the ℓ_1 -norm, the approach was transferred to the difference of a diagonal and a rank-1 matrix in [6]. A systematic analysis for both cases where a rank- r modification is allowed, is presented in this paper.

3. Proximal calculus in \mathcal{H}_V . A key step for efficiently implementing Algorithm 1 is the evaluation of the proximity operator in (2.3). Even if the proximal mapping prox_h can be computed efficiently, in general, this is not true for prox_h^V . However, we construct V of the form “diagonal \pm rank r ”, for which we propose an efficient calculus in this section. In order to cover this topic broadly, we assume $V = P \pm Q$ is a rank- r modification Q of a matrix P . The main result (Theorem 3.4) shows that the proximity operator prox_h^V in the modified metric V can be reduced essentially to the proximity operator prox_h^P without the rank- r modification and an r -dimensional root finding problem.

3.1. Preliminaries. We only recall here essential definitions. More notions, results from convex analysis as well as proofs are deferred to the appendix.

DEFINITION 3.1 (Proximity operator [48]). *Let $h \in \Gamma_0(\mathcal{H})$. Then, for every $x \in \mathcal{H}$, the function $z \mapsto \frac{1}{2}\|x - z\|^2 + h(z)$ achieves its infimum at a unique point denoted by $\text{prox}_h(x)$. The single-valued operator $\text{prox}_h : \mathcal{H} \rightarrow \mathcal{H}$ thus defined is the proximity operator or proximal mapping of h . Equivalently, $\text{prox}_h = (\text{Id} + \partial h)^{-1}$ where ∂h is the subdifferential of h .*

Throughout, we denote by

$$\text{prox}_h^V(x) = \underset{z \in \mathcal{H}}{\text{argmin}} h(z) + \frac{1}{2}\|x - z\|_V^2 = (\text{Id} + V^{-1}\partial h)^{-1}(x) , \quad (3.1)$$

the proximity operator of h w.r.t. the norm endowing \mathcal{H}_V for some $V \in \mathbb{S}_{++}(N)$. Note that since $V \in \mathbb{S}_{++}(N)$, the proximity operator prox_h^V is well-defined. The proximity operator prox_h^V can also be expressed in the metric of \mathcal{H} .

LEMMA 3.2. *Let $h \in \Gamma_0(\mathcal{H})$ and $V \in \mathbb{S}_{++}(N)$. Then, the following holds:*

$$\text{prox}_h^V(x) = V^{-1/2} \circ \text{prox}_{h \circ V^{-1/2}} \circ V^{1/2}(x) .$$

The proof is in Section B.1. The important Moreau identity can be translated to the space \mathcal{H}_V .

LEMMA 3.3 (Moreau identity in \mathcal{H}_V). *Let $h \in \Gamma_0(\mathcal{H})$, then for any $x \in \mathcal{H}$*

$$\text{prox}_{\rho h^*}^V(x) + \rho V^{-1} \circ \text{prox}_{h/\rho}^{V^{-1}} \circ V(x/\rho) = x, \forall 0 < \rho < +\infty . \quad (3.2)$$

For $\rho = 1$, it simplifies to

$$\text{prox}_h^V(x) = x - V^{-1} \circ \text{prox}_{h^*}^{V^{-1}} \circ V(x) . \quad (3.3)$$

The proof is in Section B.2.

3.2. Rank- r modified metric. In this section, we present the general result for a metric $V = P \pm Q \in \mathbb{S}_{++}(N)$, where $P \in \mathbb{S}_{++}(N)$ and $Q = \sum_{i=1}^r u_i u_i^\top \in \mathbb{R}^{N \times N}$ is symmetric with $\text{rank}(Q) = r$ and $r \leq N$, given by r linearly independent vectors $u_1, \dots, u_r \in \mathcal{H}$. Computing the proximity operator prox_h^V can be reduced to the simpler problem of evaluating prox_h^P and an r dimensional root finding problem, which can be solved exactly (see Section 3.3 for examples) or by generalized Newton iterations (see Section 3.2.1).

THEOREM 3.4 (Proximity operator for a rank- r modified metric). *Let $h \in \Gamma_0(\mathcal{H})$. Then,*

$$\begin{aligned} \text{prox}_h^V(x) &= P^{-1/2} \circ \text{prox}_{h \circ P^{-1/2}} \circ P^{1/2}(x \mp P^{-1}U\alpha^*) \\ &= \text{prox}_h^P(x \mp P^{-1}U\alpha^*) , \end{aligned} \quad (3.4)$$

where $U = (u_1, \dots, u_r)$ and $\alpha^* \in \mathbb{R}^r$ is the unique root of

$$\begin{aligned} l(\alpha) &= U^\top \left(x - P^{-1/2} \circ \text{prox}_{h \circ P^{-1/2}} \circ P^{1/2}(x \mp P^{-1}U\alpha) \right) + X\alpha \\ &= U^\top \left(x - \text{prox}_h^P(x \mp P^{-1}U\alpha) \right) + X\alpha , \end{aligned} \quad (3.5)$$

where $X = U^\top Q U \in \mathbb{S}_{++}(r)$. The mapping $l : \mathbb{R}^r \mapsto \mathbb{R}^r$ is Lipschitz continuous with Lipschitz constant $\|X\| + \|P^{-1/2}U\|^2$ and strongly monotone. The proof is in Section B.3.

REMARK 3.5.

- The root finding problem in Theorem 3.4 emerges from the dual problem for solving prox_h^V . Passing to the dual problem reduces dramatically the dimensionality of the problem to be solved from N to r where usually $r \ll N$. The dual problem boils down to an r -dimensional root finding problem of a strongly monotone function.
- Theorem 3.4 simplifies the computation of prox_h^V to prox_h^P (respectively, $\text{prox}_{h \circ P^{-1/2}}$), which might be much easier to solve, e.g. when P is a diagonal matrix as considered in this paper. This idea will be made more concrete in Sections 3.3.
- The variable metric forward-backward splitting algorithm requires the inverse of the metric in the forward step. It can be computed using the Sherman-Morrison inversion lemma: If $V = P \pm Q$ with $\text{rank}(Q) = r$, then

$$V^{-1} = P^{-1} \mp \tilde{Q}^{-1}, \quad \tilde{Q}^{-1} := P^{-1}Q(\text{Id} \pm P^{-1}Q)^{-1}P^{-1},$$

with $\text{rank}(\tilde{Q}^{-1}) = r$. Note that the sign of the rank- r part flips, see also Remark 2.1.

- Using the inversion formula for $V = P \pm Q$ as in the preceding item, and using Lemma 3.3 (Moreau identity in \mathcal{H}_V), the computation of the proximity operator of the convex conjugate function h^* , $\text{prox}_{h^*}^V$, can be cast in terms of computing $\text{prox}_h^{V^{-1}}$.

COROLLARY 3.6. *Let $V = P + Q_1 - Q_2 \in \mathbb{S}_{++}(N)$ with $P \in \mathbb{S}_{++}(N)$ and symmetric positive semi-definite matrices Q_1, Q_2 with $\text{rank}(Q_i) = r_i$ and let $\text{Im}(Q_i)$ be spanned by the columns of $U_i \in \mathbb{R}^{N \times r_i}$, $i = 1, 2$. Set $P_1 = P + Q_1$. Then, for $h \in \Gamma_0(\mathcal{H})$, the following holds:*

$$\text{prox}_h^V(x) = \text{prox}_{h_1}^{P_1}(x + P_1^{-1}U_1\alpha_1^*) = \text{prox}_h^P(x + P_1^{-1}U_1\alpha_1^* - P^{-1}U_2\alpha_2^*)$$

where $\alpha_i^* \in \mathbb{R}^{r_i}$, $i = 1, 2$, are the unique roots of the coupled system

$$\begin{aligned} l_1(\alpha_1, \alpha_2) &= U_1^\top (x - \text{prox}_h^P(x + P_1^{-1}U_1\alpha_1 - P^{-1}U_2\alpha_2)) - X_1\alpha_1 \\ l_2(\alpha_1, \alpha_2) &= U_2^\top (x + P_1^{-1}U_1\alpha_1 - \text{prox}_h^P(x + P_1^{-1}U_1\alpha_1 - P^{-1}U_2\alpha_2)) - X_2\alpha_2 \end{aligned}$$

where $X_i = U_i^\top Q_i U_i \in \mathbb{S}_{++}(N)$, $i = 1, 2$.

Proof. Corollary 3.6 follows from a recursive application of Theorem 3.4 to prox_h^V with $V = P_1 - Q_2$ and prox_h^P with $P_1 = P + Q_1$. \square

3.2.1. Semi-smooth Newton method. As discussed above, depending on the structure of the proximity operator $\text{prox}_{h \circ P^{-1/2}}$, either general-purpose or specialized algorithms for solving the root-finding problem can be derived. In some situations, see e.g. Proposition 3.11, the root of the function l can be found exactly in linear time. If no special structure is available, however, one can appeal to some efficient iterative method to solve (3.5). As l is Lipschitz-continuous, and thus *Newton* or *slantly* differentiable, semi-smooth Newton methods (see e.g. [49, 50]) are good solvers, with the proviso that one can design a simple slanting function which can be algorithmically exploited.

The semi-smooth Newton method in [50] for the solution of (3.5) can be stated as the iteration

$$\alpha_{n+1} = \alpha_n - g(\alpha_n)^{-1} l(\alpha_n) , \quad (3.6)$$

where g is a generalized derivative of l .

PROPOSITION 3.7 (Generalized derivative for the root finding problem). *Consider the situation of Theorem 3.4. If $\text{prox}_{h \circ P^{-1/2}}$ is Newton differentiable with generalized derivative G , then so is the mapping l with a generalized derivative $g : \mathbb{R}^r \mapsto \mathbb{R}^{r \times r}$*

$$g(\alpha) = U^\top (Q^\pm \pm P^{-1/2} \circ G(P^{1/2}x \mp P^{-1/2}U\alpha) \circ P^{-1/2})U .$$

Furthermore, g is nonsingular with a uniformly bounded inverse on \mathbb{R} . The proof is in Section B.4.

Thus, as l is Newton differentiable with nonsingular generalized derivative whose inverse is also bounded, the general semi-smooth Newton convergence theorem (e.g. [50, Theorem 3.13]) implies that (3.6) converges super-linearly to the unique root of (3.8).

3.3. Diagonal \pm rank-1 metric. Here we deal with metrics of the form $V = D \pm uu^\top \in \mathbb{S}_{++}(N)$ which will be at the heart of our quasi-Newton splitting algorithm, where D is diagonal with (strictly) positive diagonal elements d_i , and $u \in \mathbb{R}^N$.

3.3.1. General case. THEOREM 3.8 (Proximity operator for a diagonal \pm rank-1 metric). *Let $h \in \Gamma_0(\mathcal{H})$. Then,*

$$\text{prox}_h^V(x) = D^{-1/2} \circ \text{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x \mp \alpha^* D^{-1}u) , \quad (3.7)$$

where α^* is the unique root of

$$l(\alpha) = \left\langle u, x - D^{-1/2} \circ \text{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x \mp \alpha D^{-1}u) \right\rangle + \alpha , \quad (3.8)$$

which is a strongly increasing and Lipschitz continuous function on \mathbb{R} with Lipschitz constant $1 + \sum_i u_i^2/d_i$.

Theorem 3.8 is a specialization of Theorem 3.4.

REMARK 3.9.

- It is of course straightforward to compute prox_h^V from $\text{prox}_h^{V^{-1}}$ either using Theorem 3.8, or using this theorem together with Lemma 3.3 and the Sherman-Morrison inversion lemma. Indeed, when $V = D \pm uu^\top$ then $V^{-1} = D^{-1} \mp vv^\top$, where $v = D^{-1}u / \sqrt{1 \pm \sum_i \frac{u_i^2}{d_i}}$.
- The formula for the inverse is also important for the forward step (2.3) in Algorithm 2.
- The theory developed in [2] accounts for the proximity operator w.r.t. a metric $V = D + uu^\top$ (diagonal + rank-1), which is extended here to the case $V = D \pm uu^\top$.

Karimi and Vavasis [6] developed an algorithm for solving the proximity operator of the (separable) ℓ_1 -norm with respect to a metric $V = D - uu^\top$, which is not covered in [2]. The results in Theorems 3.4 and 3.8 generalize and formalize the algorithmic procedure in [6].

3.3.2. Separable case. The following corollary states that the proximity operator takes an even more convenient form when h is separable. It is a specialization of Theorem 3.8.

COROLLARY 3.10 (Proximity operator for a diagonal \pm rank-1 metric for separable functions). *Assume that $h \in \Gamma_0(\mathcal{H})$ is separable, i.e. $h(x) = \sum_{i=1}^N h_i(x_i)$, and $V = D \pm uu^\top \in \mathbb{S}_{++}(N)$, where D is diagonal with (strictly) positive diagonal elements d_i , and $u \in \mathbb{R}^N$. Then*

$$\text{prox}_h^V(x) = \left(\text{prox}_{h_i/d_i}(x_i \mp \alpha^* u_i/d_i) \right)_{i=1}^N, \quad (3.9)$$

where α^* is the unique root of

$$l(\alpha) = \left\langle u, x - \left(\text{prox}_{h_i/d_i}(x_i \mp \alpha u_i/d_i) \right)_{i=1}^N \right\rangle + \alpha, \quad (3.10)$$

which is a Lipschitz continuous and strongly increasing function on \mathbb{R} .

In particular, when the proximity operator of each h_i is piecewise affine, we get the following.

PROPOSITION 3.11. *Consider the situation of Corollary 3.10. Assume that for $1 \leq i \leq N$, prox_{h_i/d_i} is piecewise affine on \mathbb{R} with $k_i \geq 1$ segments, i.e.*

$$\text{prox}_{h_i/d_i}(x_i) = \begin{cases} a_0^i x_i + b_0^i, & \text{if } x_i \leq t_1^i; \\ a_j^i x_i + b_j^i, & \text{if } t_j^i \leq x_i \leq t_{j+1}^i, \quad j \in \{1, \dots, k_i\}; \\ a_{k_i+1}^i x_i + b_{k_i+1}^i, & \text{if } t_{k_i+1}^i \leq x_i, \end{cases}$$

for some $a_j^i, b_j^i \in \mathbb{R}$, and define $t_0^i := -\infty$ and $t_{k_i+2}^i := +\infty$. Then $\text{prox}_h^V(x)$ can be obtained exactly using Algorithm 4 with binary search for Step 3 in $O(K \log(K))$ steps where $K = \sum_{i=1}^N k_i \geq N$. The proof is in Section B.5.

Using Proposition 3.11, we derive Algorithm 4.

Algorithm 4 Exact root finding algorithm for piecewise linear separable proximity operators

Require: Piecewise linear proximity operator $\text{prox}_{h_i/d_i}(x_i)$, $i = 1, \dots, N$, as defined in Prop. 3.11.

- 1: Sort $\bar{\theta} := \bigcup_{i=1}^N \{\pm \frac{d_i}{u_i}(x_i - t_j^i) : j = 1, \dots, k_i\} \subset \mathbb{R}$ into a list $\theta \in \mathbb{R}^{k'}$ with $k' \leq K$.
- 2: Set $\bar{\theta} := [-\infty, \theta_1, \dots, \theta_{k'}, +\infty]$.
- 3: Via the bisection method, detect the interval $[\theta_-, \theta_+)$ with adjacent $\theta_-, \theta_+ \in \bar{\theta}$ that contains the root of $l(\alpha)$.
- 4: Compute the root $\alpha^* = -b/a$ where a and b are determined as follows:
- 5: For all $i = 1, \dots, N$, define $j_i \in \{0, \dots, k_i + 1\}$ such that $t_{j_i}^i \leq \theta_- < \theta_+ \leq t_{j_i+1}^i$, and compute

$$a := 1 \pm \sum_{i=1}^N a_{j_i}^i u_i^2/d_i \quad \text{and} \quad b := \sum_{i=1}^N u_i((1 - a_{j_i}^i)x_i - b_{j_i}^i).$$

Some remarks are in order.

REMARK 3.12.

- The “ \pm ” in Algorithm 4 refers to the two cases of $V = D \pm uu^\top$ from Corollary 3.10.
- Since (3.10) is piecewise affine, a, b in Step 5 for the interval $[\theta_-, \theta_+)$, can be determined by

$$a = \frac{l(\theta'_+) - l(\theta'_-)}{\theta'_+ - \theta'_-} \quad \text{and} \quad b = l(\theta'_-)$$

where $\theta_- \leq \theta'_- < \theta'_+ \leq \theta_+$ and $-\infty < \theta'_-$ and $\theta'_+ < +\infty$. (The usage of “ θ' ” avoids “ $l(-\infty)$ ”.)

REMARK 3.13.

- The bulk of complexity in Proposition 3.11 lies in locating the appropriate breakpoints. This can be achieved straightforwardly by sorting followed by a bisection search, as advocated, whose computational cost is nearly linear in N up to a logarithmic factor. The log term can theoretically be removed by replacing sorting with a median-search-like procedure whose expected complexity is linear.
- The above computational cost can be reduced in many situations by exploiting e.g. symmetry of the h_i s, identical functions, etc. This turns out to be the case for many functions of interest, e.g. ℓ_1 -norm, indicator of the ℓ_∞ -ball or the positive orthant, polyhedral gauges, and many others; see examples hereafter.
- Corollary 3.10 can be extended to the “block” separable case (i.e. separable in subsets of coordinates) when D is constant along the same block indices.
- It is important to realize that the idea underlying Proposition 3.11 extends to the more general case where the proximity operator of each h_i is piecewise-smooth. The main difference is that in this case, once the appropriate breakpoints are located, one will have to find the zero of a one-dimensional non-linear smooth function. This can be done e.g. using Newton or even higher order method. For instance, this turns to be the case for the $\ell_1 - \ell_2$ norm which is popularly used to promote group sparsity. This example will be considered in more detail in Section 3.3.3.

3.3.3. Examples. Many functions can be handled very efficiently using our results above. For instance, Table 3.1 summarizes a few of them where we can obtain either an exact answer by sorting when possible, or else by minimizing w.r.t. to a scalar variable (i.e. finding the unique root of (3.8)).

Affine constraint. We start with a case where the proximity operator in the diagonal \pm rank 1 metric has a closed-form expression. Consider the case where $h = \iota_{\{x: Ax=b\}}$. We then immediately get

$$\text{prox}_{h \circ D^{-1/2}}(z) = z + Y^+(b - Yz) = \Pi z + c$$

where $Y = AD^{-1/2}$, Π is the projector on $\text{Ker}(Y) = D^{1/2} \text{Ker}(A)$, and $c = Y^+b$. After simple algebra, it follows from Theorem 3.8, that the unique root of l in this case is

$$\alpha^* = \frac{\langle u, D^{-1/2}(c - (\text{Id} - \Pi)D^{1/2}x) \rangle}{1 \pm \langle u, D^{-1/2}\Pi D^{-1/2}u \rangle}.$$

Positive orthant. We now put Proposition 3.11 on a more concrete footing by explicitly covering the case when h represents non-negativity constraints. Consider $V = D + uu^\top$ and $h = \iota_{\{x: x \geq 0\}}$. We will calculate

$$\text{prox}_h^{V^{-1}}(x) = \underset{y \geq 0}{\text{argmin}} \frac{1}{2} \|y - x\|_{V^{-1}}^2 \quad (3.11)$$

Function h	Method
ℓ_1 -norm (separable)	exact with sorting
Hinge (separable)	exact with sorting
Box constraint (separable)	exact with sorting
ℓ_∞ -ball (separable)	exact with sorting
Positivity constraint (separable)	exact with sorting
$\ell_1 - \ell_2$ (block-separable)	sort and root finding
Affine constraint (nonseparable)	closed-form
ℓ_1 -ball (nonseparable)	root-finding and $\text{prox}_{h \circ D^{-1/2}}$ costs a sort
ℓ_∞ -norm (nonseparable)	from projector on the ℓ_1 -ball by Moreau-identity
Simplex (nonseparable)	root-finding and $\text{prox}_{h \circ D^{-1/2}}$ costs a sort
max function (nonseparable)	from projector on the simplex by Moreau-identity

Table 3.1: A few examples of functions which have efficiently computable proximity operators in the metric $V = D \pm uu^\top$.

We use the fact that the projector on the positive orthant is separable with components $(x_i)_+ := \max(0, x_i)$, i.e. a piecewise affine function. Define the scalar $\alpha = u^\top \lambda$. Let $\lambda_i^{(\alpha)} := (-(x_i + \alpha u_i)/d_i)_+$, so we search for a value of α such that $\alpha = u^\top \lambda^{(\alpha)}$, or in other words, a root of $l(\alpha) = \alpha - u^\top \lambda^{(\alpha)}$.

Define $\hat{\alpha}_i$ to be the sorted values of $(-x_i/u_i)$, so we see that l is linear in the regions $[\hat{\alpha}_i, \hat{\alpha}_{i+1}]$ and so it is trivial to check if l has a root in this region. Thus the problem is reduced to finding the correct region i , which can be done efficiently by a bisection search over $\log_2(n)$ values of i since l is monotonic. To see that l is monotonic, we write it as

$$l(\alpha) = \alpha + \sum_{i=1}^N ((u_i x_i + \alpha u_i^2)/d_i) \chi_i(\alpha)$$

where $\chi_i(\alpha)$ encodes the positivity constraint in the argument of $(\cdot)_+$ and is thus either 0 or 1, hence the slope is always positive.

$\ell_1 - \ell_2$ norm. Let \mathcal{B} be a uniform disjoint partition of $\{1, \dots, N\}$, i.e. $\bigcup_{b \in \mathcal{B}} = \{1, \dots, n\}$ and $b \cap b' = \emptyset$ for all $b \neq b' \in \mathcal{B}$. The $\ell_1 - \ell_2$ norm of x is

$$\|x\|_{1,2} = \sum_{b \in \mathcal{B}} \|x_b\| \quad (3.12)$$

where x_b is the subvector of x indexed by block b .

Without loss of generality, we assume that all blocks have the same size, and we consider the metric $V = D + uu^\top$, where the diagonal matrix D is constant on each block b . We now detail how to compute the proximity operator in \mathcal{H}_V of $h = \lambda \|\cdot\|_{1,2}$, $\lambda > 0$. For this, we will exploit Theorem 3.8 and the expression of $\text{prox}_{h \circ D^{-1/2}}$, i.e. block soft-thresholding. The latter gives

$$\left(D^{-1/2} \text{prox}_{h \circ D^{-1/2}}(D^{1/2}x) \right)_b = (\text{prox}_{h \circ D^{-1}}(x))_b = \left(1 - \frac{\lambda}{d_b \|x_b\|} \right)_+ x_b, \quad \forall b \in \mathcal{B},$$

where d_b is the diagonal entry of D shared by block b . This then entails that

$$l(\alpha) = \langle x, u \rangle + \alpha - \sum_{b \in \mathcal{J}(\alpha)} \left(\left(1 - \frac{\lambda}{d_b \|x_b - \alpha u_b/d_b\|} \right) (\langle x_b, u_b \rangle - \alpha \|u_b\|^2/d_b) \right),$$

where $\mathcal{I}(\alpha) = \{b \in \mathcal{B} : \|x_b - \alpha u_b / d_b\| \geq \lambda / d_b\}$. This is a piecewise smooth function, with breakpoints at the values of α where the active support $\mathcal{I}(\alpha)$ changes. To compute the root of α , it is sufficient to locate the two breakpoints where l changes sign, and then run a fast root-finding algorithm (*e.g.* Newton’s method) on this interval where α is actually C^∞ . Denote $N_{\mathcal{B}} = \lfloor N/|b| \rfloor$ the number of blocks. There are at most $2N_{\mathcal{B}}$ breakpoints, and these correspond to the two real roots of $N_{\mathcal{B}}$ univariate quadratic polynomials, each corresponding to

$$\|d_b x_b - \alpha u_b\|^2 = \alpha^2 \|u_b\|^2 - 2\alpha d_b \langle x_b, u_b \rangle + d_b^2 \|x_b\|^2 = \lambda^2 .$$

Sorting these roots costs at most $O(N_{\mathcal{B}} \log N_{\mathcal{B}})$. To locate the breakpoints, a simple procedure is a bisection search on the sorted values, and each step necessitates to evaluate l . This search also costs at most $O(N_{\mathcal{B}} \log N_{\mathcal{B}})$ operations (observe that all inner products and norms in l can be computed once for all). In summary, locating the interval of breakpoints containing the root takes $O(N_{\mathcal{B}} \log N_{\mathcal{B}})$ operations, though we believe this complexity could be made linear in $N_{\mathcal{B}}$ with an extra effort.

4. A SR1 forward–backward algorithm.

4.1. Metric construction. Following the conventional quasi-Newton notation, we let B denote an approximation to the Hessian of f and H denote an approximation to the inverse Hessian. All quasi-Newton methods update an approximation to the (inverse) Hessian that satisfies the *secant condition*:

$$H_k y_k = s_k, \quad \text{where} \quad y_k = \nabla f(x_k) - \nabla f(x_{k-1}), \quad s_k = x_k - x_{k-1}. \quad (4.1)$$

Algorithm 2 follows the SR1 method [51], which uses a rank-1 update to the inverse Hessian approximation at every step. The SR1 method is perhaps less well-known than BFGS, but it has the crucial property that updates are rank-1, rather than rank-2, and it is described “[SR1] has now taken its place alongside the BFGS method as the pre-eminent updating formula.” [52].

We propose two important modifications to SR1. The first is to use limited-memory, as is commonly done with BFGS. In particular, we use zero-memory, which means that at every iteration, a new diagonal plus rank-one matrix is formed. The other modification is to extend the SR1 method to the general setting of minimizing $f + h$ where f is smooth but h need not be smooth; this further generalizes the case when h is an indicator function of a convex set. Every step of the algorithm replaces f with a quadratic approximation, and keeps h unchanged. Because h is left unchanged, the subgradient of h is used in an *implicit* manner, in comparison to methods such as [28] that use an approximation to h as well and therefore take an *explicit* subgradient step.

Choosing H_0 . In our experience, the choice of H_0 is best if scaled with a Barzilai–Borwein spectral step length

$$\tau_{\text{BB2}} = \langle s_k, y_k \rangle / \langle y_k, y_k \rangle \quad (4.2)$$

(we call it τ_{BB2} to distinguish it from the other Barzilai–Borwein step size $\tau_{\text{BB1}} = \langle s_k, s_k \rangle / \langle s_k, y_k \rangle \geq \tau_{\text{BB2}}$).

In SR1 methods, the quantity $\langle s_k - H_0 y_k, y_k \rangle$ must be positive in order to have a well-defined update for u_k . The update is:

$$H_k = H_0 + u_k u_k^\top, \quad u_k = (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle}. \quad (4.3)$$

For this reason, we choose $H_0 = \gamma \tau_{\text{BB2}} \text{Id}$ with $0 < \gamma < 1$, and thus $0 \leq \langle s_k - H_0 y_k, y_k \rangle = (1 - \gamma) \langle s_k, y_k \rangle$. If $\langle s_k, y_k \rangle = 0$, then there is no symmetric rank-one update that satisfies

Algorithm 5 Sub-routine to compute the approximate inverse Hessian H_k , 0SR1 variant

Require: k, s_k, y_k as in (4.1); and $0 < \gamma < 1$, $0 < \tau_{\min} < \tau_{\max}$

```

1: if  $k = 1$  then
2:    $H_0 \leftarrow \tau \text{Id}$  where  $\tau > 0$  is arbitrary
3:    $u_k \leftarrow 0$ 
4: else
5:    $\tau_{\text{BB2}} \leftarrow \frac{\langle s_k, y_k \rangle}{\|y_k\|^2}$  {Barzilai–Borwein step length}
6:   Project  $\tau_{\text{BB2}}$  onto  $[\tau_{\min}, \tau_{\max}]$ 
7:    $H_0 \leftarrow \gamma \tau_{\text{BB2}} \text{Id}$ 
8:   if  $\langle s_k - H_0 y_k, y_k \rangle \leq 10^{-8} \|y_k\|_2 \|s_k - H_0 y_k\|_2$  then
9:      $u_k \leftarrow 0$  {Skip the quasi-Newton update}
10:  else
11:     $u_k \leftarrow (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle}$ .
12:  end if
13: end if
14: return  $H_k = H_0 + u_k u_k^\top$  { $B_k = H_k^{-1}$  can be computed via the Sherman-Morrison formula}

```

the secant condition. The inequality $\langle s_k, y_k \rangle > 0$ is the *curvature condition*, and it is guaranteed for all strictly convex objectives. Following the recommendation in [53], we skip updates whenever $\langle s_k, y_k \rangle$ cannot be guaranteed to be non-zero given standard floating-point precision.

A value of $\gamma = 0.8$ works well in most situations. We have tested picking γ adaptively, as well as trying H_0 to be non-constant on the diagonal, but found no consistent improvements.

4.2. Convergence analysis. For our convergence analysis, we naturally assume that f is also μ -strongly convex. The following lemma delivers useful uniform bounds on the eigenvalues of matrices H_k .

LEMMA 4.1. *Suppose that f is μ -strongly convex and its gradient is L -Lipschitz. Then, $\forall k \geq 0$, $a \text{Id} \preceq H_k \preceq b \text{Id}$, $0 < a = \gamma L^{-1}$, $0 < b = \frac{(1+\gamma)\mu^{-1} - 2\gamma L^{-1}}{1-\gamma}$. The proof is in Section C.1.*

THEOREM 4.2. *Suppose that f is μ -strongly convex and its gradient is L -Lipschitz. Let a and b be given as in Lemma 4.1. Assume that $0 < \underline{\kappa} \leq \kappa_k \leq \bar{\kappa} < 2(Lb)^{-1}$. Let $\alpha = 1 - \frac{Lb\bar{\kappa}}{2}$ and $\eta = \frac{L}{2\gamma\mu\underline{\kappa}}$. Then, the sequence of iterates $(x_k)_{k \in \mathbb{N}}$ of the 0SR1 forward-backward Algorithm 2 with $t = 1$ converge linearly to the unique minimizer x^* , i.e.*

$$\|x_k - x^*\| \leq \sqrt{\frac{2(F(x_0) - F(x^*))}{\mu}} \rho^{k/2},$$

where

$$\rho = \begin{cases} \rho_1 & \text{for } \alpha \in]0, 1/2[\\ \min(\rho_1, \rho_2) & \text{for } \alpha \in [1/2, 1[\end{cases},$$

with

$$\rho_1 = 1 - \alpha \left(1 - 2 \left(\sqrt{\eta^2 + \eta} - \eta \right) \right) \quad \text{and} \quad \rho_2 = \begin{cases} 2\eta (\leq 1/2) & \text{if } \eta \leq 1/4 \\ 1 - \frac{1}{8\eta} & \text{otherwise} \end{cases}.$$

The proof is in Section C.2. Fig. 4.1 shows the phase diagram of the rate ρ as a function of η and α .

REMARK 4.3. For a concrete example of the rates in Theorem 4.2, choose $\gamma = 1/2$ so that $a = 1/(2L)$ and $b = 3\mu^{-1} - 2L^{-1}$, and choose $\kappa_k \equiv \bar{\kappa} = \underline{\kappa} = 1/(Lb)$. Thus $\alpha = 1/2$. Let $c = L/\mu$ be the condition number of the problem. Then $\eta = c(3c - 2)$, and so for large $c \gg 1$, we have $\eta \gg 1$ and via Taylor expansion we see that $\rho_1 - \rho_2 \rightarrow 0$ as $\eta \rightarrow \infty$. In turn, the rate of linear convergence is $\rho \approx 1 - 1/(8\eta) \approx 1 - 1/(24c^2)$.

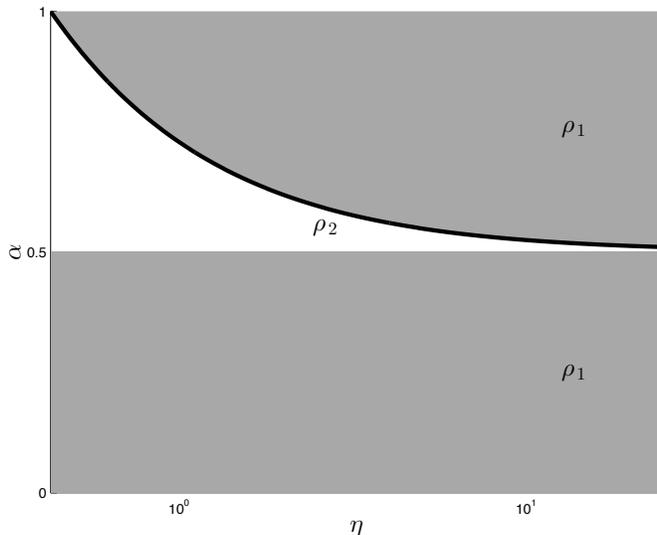


Figure 4.1: Convergence rate as a function of the parameters η and α (see Theorem 4.2 for details).

5. L-BFGS forward–backward splitting. In this section, we show how the extended theory for rank- r modified proximity operators in Section 3.2 can be used for the efficient treatment of the more sophisticated L-BFGS method in our context of proximal quasi-Newton methods. We consider Algorithm 2 where the metric construction is outlined in Section 5.1 following the notation in [53]. The proximity operator in (2.3) will be of type “diagonal \pm rank-2”.

5.1. Metric construction. Define

$$\rho_k = \frac{1}{y_k^\top s_k}, \quad V_k = \text{Id} - \rho_k y_k s_k^\top.$$

with

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

as in (4.1). Store $\{s_i, y_i\}$ for $i = k - m, k - m - 1, \dots, k - 1$. Choose H_k^0 as before, e.g., $H_k^0 = \gamma\tau\text{Id}$. Then the limited-memory BFGS (L-BFGS) quadratic approximation is

$$\begin{aligned} H_k &= (V_{k-1}^\top \cdots V_{k-m}^\top) H_k^0 (V_{k-m} \cdots V_{k-1}) \\ &\quad + \rho_{k-m} (V_{k-1}^\top \cdots V_{k-m+1}^\top) s_{k-m} s_{k-m}^\top (V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \rho_{k-m+1} (V_{k-1}^\top \cdots V_{k-m+2}^\top) s_{k-m+1} s_{k-m+1}^\top (V_{k-m+2} \cdots V_{k-1}) \\ &\quad + \cdots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^\top. \end{aligned}$$

In the classical (unconstrained) L-FBGS, the update is then $x_{k+1} = x_k - \alpha_k H_k \nabla f_k$.

In the extreme low-memory case ($m = 1$), we have

$$H_{k+1} = V_k^\top H_k^0 V_k + \rho_k s_k s_k^\top$$

which gives us a 0-BFGS method. For this $m = 1$ case and $\tau = \tau_{\text{BB2}}$, writing V for V_k and so on, we can expand

$$\begin{aligned} H_k &= V^\top H_k^0 V + \rho s s^\top \\ &= (\text{Id} - \rho s y^\top) (\gamma\tau \text{Id}) (\text{Id} - \rho y s^\top) + \rho s s^\top \\ &= \gamma\tau (\text{Id} - \rho(y s^\top + s y^\top) + \rho^2 \|y\|^2 s s^\top) + \rho s s^\top \\ [\rho \|y\|^2 \tau = 1] &= \gamma\tau \text{Id} + \rho(1 + \gamma) \left(s s^\top - \frac{\gamma\tau}{1 + \gamma} (s y^\top + y s^\top) + \frac{\gamma^2 \tau^2}{(1 + \gamma)^2} y y^\top \right) - \rho \frac{\gamma^2 \tau^2}{1 + \gamma} y y^\top \\ &= \gamma\tau \text{Id} + \rho(1 + \gamma) \underbrace{\left(s - \frac{\gamma\tau}{1 + \gamma} y \right)}_{=: u_\gamma} \left(s - \frac{\gamma\tau}{1 + \gamma} y \right)^\top - \rho \frac{\gamma^2 \tau^2}{1 + \gamma} y y^\top, \end{aligned} \tag{5.1}$$

which shows that the inverse Hessian approximation is of type “diagonal + rank-1 – rank-1” with positive semi-definite rank-1 matrices. Note that we are free to choose $\gamma = 1$, in which case the simpler expression follows:

$$H_k = \tau \text{Id} + 2\rho \left(s - \frac{\tau}{2} y \right) \left(s - \frac{\tau}{2} y \right)^\top - \rho \frac{\tau^2}{2} y y^\top, \tag{5.2}$$

Applying the Sherman–Morrison inversion lemma to this, we obtain the following approximation to the Hessian matrix $B_k = H_k^{-1}$:

$$B_k = B_k^0 - \frac{B_k^0 s s^\top B_k^0}{s^\top B_k^0 s} + \frac{y y^\top}{y^\top s} = \frac{1}{\gamma\tau} \left(\text{Id} - \frac{s s^\top}{s^\top s} + \gamma\tau \frac{y y^\top}{y^\top s} \right) \stackrel{\tau = \tau_{\text{BB2}}}{=} \frac{1}{\gamma\tau_{\text{BB2}}} \left(\text{Id} - \frac{s s^\top}{s^\top s} + \gamma \frac{y y^\top}{y^\top y} \right).$$

The proximity operator with respect to this metric can be computed as shown in Corollary 3.6. Only the evaluation of the simple proximity operator $\text{prox}_h^{B_0}$ is required. The main computational cost comes from the two dimensional root finding problem, which can be solved efficiently using semi-smooth Newton methods.

5.2. Convergence analysis. For our convergence analysis, we naturally assume that f is also μ -strongly convex. The following lemma delivers useful uniform bounds on the eigenvalues of matrices H_k .

LEMMA 5.1. *Suppose that f is μ -strongly convex and its gradient is L -Lipschitz. Then, $\forall k \geq 0$, $a\text{Id} \preceq H_k \preceq b\text{Id}$, $0 < a = \gamma/(1 + \gamma)L^{-1}$, $0 < b = (1 + 2\gamma)\mu^{-1} - \frac{(2 + \gamma)\gamma}{1 + \gamma}L^{-1}$.*

The proof is in Section D.1.

THEOREM 5.2. *Suppose that f is μ -strongly convex and its gradient is L -Lipschitz. Let $\gamma > 0$, and a, b be given as in Lemma 5.1. Assume that $0 < \underline{\kappa} \leq \kappa_k \leq \bar{\kappa} < 2(Lb)^{-1}$. Let $\alpha = 1 - \frac{Lb\bar{\kappa}}{2}$ and $\eta = \frac{L}{2\gamma\mu\underline{\kappa}}$. Then, the sequence of iterates $(x_k)_{k \in \mathbb{N}}$ of the L-BFGS forward-backward Algorithm 3 (with H_k as in (5.1)) with $t = 1$ converges linearly to the unique minimizer x^* , i.e.*

$$\|x_k - x^*\| \leq \sqrt{\frac{2(F(x_0) - F(x^*))}{\mu}} \rho^{k/2},$$

where

$$\rho = \begin{cases} \rho_1 & \text{for } \alpha \in]0, 1/2[\\ \min(\rho_1, \rho_2) & \text{for } \alpha \in [1/2, 1[\end{cases},$$

with

$$\rho_1 = 1 - \alpha \left(1 - 2 \left(\sqrt{\eta^2 + \eta} - \eta \right) \right) \quad \text{and} \quad \rho_2 = \begin{cases} 2\eta (\leq 1/2) & \text{if } \eta \leq 1/4 \\ 1 - \frac{1}{8\eta} & \text{otherwise} \end{cases}.$$

The proof is the same as that of Theorem 4.2 (Section C.2) by substituting the constants a and b in Lemma 4.1 with those in Lemma 5.1. Note that the phase diagram in Fig. 4.1 still applies, though the underlying constants are slightly changed.

REMARK 5.3. *Let's again illustrate the rate in Theorem 5.2. We choose $\gamma = 1/2$ as in Remark 4.3 so that $a = 1/(3L)$, $b = 2\mu^{-1} - 5/6L^{-1}$, and $\kappa_k \equiv \bar{\kappa} = \underline{\kappa} = 1/(Lb)$. Thus $\alpha = 1/2$ and $\eta = c(2c - 5/6)$, where $c = L/\mu$ is the condition number of the problem. For large $c \gg 1$, the rate of linear convergence is $\rho \approx 1 - 1/(8\eta) \approx 1 - 1/(16c^2)$, which is smaller than the one of OSR1 in Remark 4.3.*

6. Numerical experiments and comparisons. In the spirit of reproducible research, and to record the exact algorithmic details, all code for experiments from this paper is available at <https://github.com/stephenbecker/zeroSR1/tree/master/paperExperiments>.

6.1. LASSO problem. Consider the unconstrained LASSO problem (1.1). Many codes, such as [54] and L-BFGS-B [3], handle only non-negativity or box-constraints. Using the standard change of variables by introducing the positive and negative parts of x , the LASSO can be recast as

$$\min_{x_+, x_- \geq 0} \frac{1}{2} \|Ax_+ - Ax_- - b\|^2 + \lambda \mathbf{1}^\top (x_+ + x_-) \quad (6.1)$$

and then x is recovered via $x = x_+ - x_-$. With such a formulation solvers such as L-BFGS-B are applicable. However, this constrained problem has twice the number of variables, and the Hessian of the quadratic part changes from $A^\top A$ to $\tilde{A} = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}$ which necessarily has (at least) n degenerate 0 eigenvalues and adversely affects solvers.

A similar situation occurs with the hinge-loss function. Consider the shifted and reversed hinge loss function $h(x) = \max(0, x)$. Then one can split $x = x_+ - x_-$, add constraints $x_+ \geq 0, x_- \geq 0$, and replace $h(x)$ with $\mathbf{1}^\top (x_+)$. As before, the Hessian gains n degenerate eigenvalues.

We compared our proposed algorithm on the LASSO problem. The first example, on the left of Figure 6.1, is a typical example from compressed sensing that takes $A \in \mathbb{R}^{m \times n}$

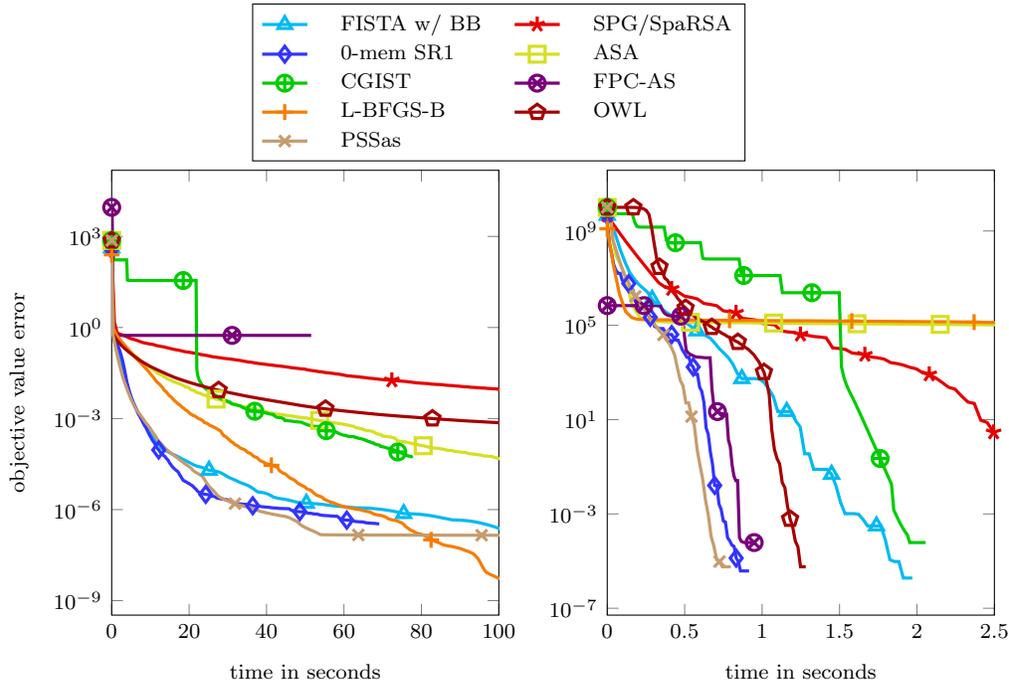


Figure 6.1: Convergence plots for the methods described in Section 6 for solving the ℓ_1 LASSO problem. The plot on the left corresponds to the experiment with the random matrix and the right plot to the experiment with the differential operator. The vertical axis is the same for both plots. The proposed 0-mem SR1 method and PSSas efficiently solves both problems. While our method generalizes easily to the $\ell_1 - \ell_2$ sparsity norm, PSSas is hard to generalize.

to have iid $\mathcal{N}(0, 1)$ entries with $m = 1500$ and $n = 3000$. We set $\lambda = 0.1$. L-BFGS-B does very well, followed closely by our proposed SR1 algorithm, PSSas, and FISTA. Note that L-BFGS-B and ASA are in Fortran and C, respectively (the other algorithms are in Matlab).

Our second example uses a square operator A with dimensions $n = 15^3 = 3375$ chosen as a 3D discrete differential operator. This example stems from a numerical analysis problem to solve a discretized PDE as suggested by [57]. For this example, we set $\lambda = 1$. For all the solvers, we use the same parameters as in the previous example. Unlike the previous example, the right of Figure 6.1 now shows that L-BFGS-B is very slow on this problem. The FPC-AS method, very slow on the earlier test, is now the fastest. However, just as before, our SR1 method is nearly as good as the best algorithm. FISTA is significantly outperformed by our method on this problem. This robustness is one benefit of our approach, since the method does not rely on active-set identifying parameters and inner iteration tolerances. Moreover, the proposed SR1 method easily generalizes to other regularization terms.

6.2. Group LASSO problem. As a second experiment, we replace the ℓ_1 sparsity term $\|x\|_1$ in (1.1) with an $\ell_1 - \ell_2$ sparsity $\|x\|_{2,1}$ as in (3.12), which is known to promote group sparsity (hence the name group LASSO). We partition the N coordinates of $x \in \mathbb{R}^N$ into groups $b \in \mathcal{B}$ with randomly selected size $|b| \leq 12$. For the numerical experiment, the entries of A and b are drawn uniformly in $[0, 1]$, and we set $N = 2500$, $M = 1600$,

Acronym	Algorithm Name	Tests	Comments
FISTA	Fast IST Algorithm	§6.1,6.2	our own implementation in Matlab
SPG/SpaRSA	Spectral Projected Gradient[8] as used in [9]	§6.1,6.2	Matlab version from [9]
L-BFGS-B	Limited memory, box-constrained BFGS[3, 21]	§6.1	Fortran with Matlab wrapper
ASA	“Active Set Algorithm” (conjugate gradient) [23]	§6.1	C with Matlab wrapper, ver2.2
OWL	Orthant-wise Learning [24]	§6.1	Active set; Matlab
PSSas	Projected Scaled Sub-gradient + Active Set [25]	§6.1	Matlab
CGIST	“CG + IST” [27]	§6.1	Matlab
FPC-AS	“Fixed-point continuation + Active Set” [26]	§6.1	Matlab, ver. 1.21
0-mem SR1	Algorithm 5	§6.1,6.2	our approach (in Matlab)
mfZeroSR1	monotone FISTA based acceleration	§6.2	from [16], based on Algorithm 4
TsengZeroSR1	acceleration based on [55]	§6.2	from [16], based on Algorithm 4

Table 6.1: Algorithms used in experiments of sections 6.1 and 6.2. The first two algorithms are standard “first-order” algorithms; the next group of algorithms use active-set strategies; and the final group of three algorithms use a diagonal \pm rank-1 proximal mapping. Our implementation of FISTA used the Barzilai-Borwein stepsize [10] and line search, and restarted the momentum term every 1000 iterations [15]. L-BFGS-B and ASA use the reformulation of (6.1). For L-BFSG-B, we use the updated version [22]. Code for PSSas and OWL (slight variant of [24]) from [56].

and $\lambda = 1$. As the $\ell_1 - \ell_2$ norm is not polyhedral, active set based methods are hard to use. Also L-BFGS-B cannot be used, as the “trick” for the ℓ_1 -norm above does not apply here. The emerging rank-1 proximal mapping in our proposed proximal SR1 method can be solved efficiently as described in Section 3.3.3. We apply Newton’s method in the interval between breakpoints that locates the root. Besides our 0SR1 method, we test the methods mfZeroSR1 (monotone FISTA based acceleration) and TsengZeroSR1 (acceleration based on a method by Tseng [55]) that were recently proposed in [16]. These methods are also based on the diagonal \pm rank-1 proximal mapping proposed in this paper, which they implemented into an acceleration strategy similar to FISTA.

Figure 6.2 shows the convergence of several methods in terms of objective value error vs iteration (left plot) or time (right plot). Our 0SR1 method shows good performance in the low or medium precision regime. In the beginning, the SR1 metric reflects the conditioning of the problem better than isotropic metrics. In order to obtain a high precision, acceleration strategies seem to be necessary. Presumably, this comes from the $\ell_2 - \ell_1$ norm, which usually activates a whole block of coordinates, unlike in the LASSO case where eventually only a few coordinates are active and thus often has an improved condition number when restricted to these active variables. Acceleration strategies seem to compensate for this effect. The methods based on the diagonal \pm rank-1 proximal mapping outperform the state-of-the-art method FISTA. Figure 6.2 also suggests that this difference could be further increased when a more efficient implementation of the diagonal \pm rank-1 proximal mapping is used.

7. Conclusions. In this paper, we proposed a novel framework for variable metric (quasi-Newton) forward-backward splitting algorithms, designed to efficiently solve non-smooth convex problems structured as the sum of a smooth term and a non-smooth one. We introduced a class of weighted norms induced by diagonal \pm rank r symmetric positive definite matrices, as well as a calculus to compute the proximity operator in the corresponding induced metrics. The latter result is new and generalized our previous results on the subject [2], and we believe it is of independent interest as even the simpler version from

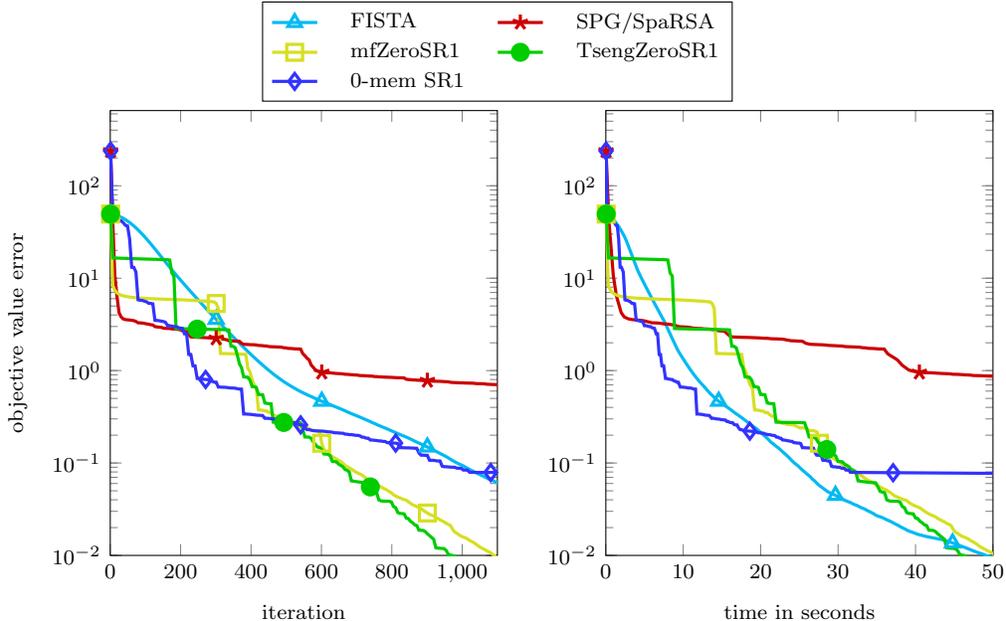


Figure 6.2: Convergence plots for the methods described in Section 6.2 for solving the $\ell_1 - \ell_2$ LASSO problem. The vertical axis is the same for both plots. The methods based on the efficient solution of the diagonal \pm rank-1 proximal mapping proposed in this paper outperform comparable methods based on a diagonally scaled proximal mapping.

[2] has been the basis of other works such as [6, 16]. We also established convergence of the algorithm, and provided clear evidence that the non-diagonal term provides significant acceleration over diagonal matrices.

The proposed method can be extended in several ways. Although we focused on forward-backward splitting, our approach can be easily extended to the new *generalized* forward-backward algorithm of [58]. However, if we switch to a primal-dual setting, which is desirable because it can handle more complicated objective functionals, updating B_k is non-obvious, though one could perhaps use our results for a non-diagonal pre-conditioning method.

Another improvement would be to derive efficient calculation for exact calculation of rank-2 proximity terms, thus allowing our 0-memory BFGS method to have cheaper and more exact update steps (as compared to the semi-smooth Newton method currently suggested). Theorem 3.4 and Corollary 3.6 give some clues in this direction.

A final possible extension is to take B_k to be diagonal plus rank-1 on diagonal blocks, since if h is separable, this is still can be solved by our algorithm (see Proposition 3.11). The challenge here is adapting this to a robust quasi-Newton update. For some matrices that are well-approximated by low-rank blocks, such as H-matrices [59], it may be possible to choose $B_k \equiv B$ to be a fixed preconditioner.

Appendix A. Elements from convex analysis. We here collect some results from convex analysis that are key for our proof. Some lemmata are listed without proof and can be either easily proved or found in standard references such as [60, 1].

A.1. Background.

Functions.

DEFINITION A.1 (Indicator function). *Let \mathcal{C} a nonempty subset of \mathcal{H} . The indicator function $\iota_{\mathcal{C}}$ of \mathcal{C} is*

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{C} , \\ +\infty, & \text{otherwise.} \end{cases}$$

$\text{dom}(\iota_{\mathcal{C}}) = \mathcal{C}$.

DEFINITION A.2 (Infimal convolution). *Let h_1 and h_2 two functions from \mathcal{H} to $\mathbb{R} \cup \{+\infty\}$. Their infimal convolution is the function from \mathcal{H} to $\mathbb{R} \cup \{\pm\infty\}$ defined by:*

$$(h_1 \overset{\dagger}{\vee} h_2)(x) = \inf \{h_1(x_1) + h_2(x_2) : x_1 + x_2 = x\} = \inf_{y \in \mathcal{H}} h_1(y) + h_2(x - y) .$$

Conjugacy.

DEFINITION A.3 (Conjugate). *Let $h : \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ having a minorizing affine function. The conjugate or Legendre-Fenchel transform of h on \mathcal{H} is the function h^* defined by*

$$h^*(v) = \sup_{x \in \text{dom}(h)} \langle v, x \rangle - h(x) .$$

LEMMA A.4 (Calculus rules).

- (i) $(h(x) + t)^*(v) = h^*(v) - t$.
- (ii) $(th(x))^*(v) = tf^*(v/t)$, $t > 0$.
- (iii) $(h \circ A)^* = h^* \circ (A^{-1})^*$ if A is a linear invertible operator.
- (iv) $(h(x - x_0))^*(v) = h^*(v) + \langle v, x_0 \rangle$.
- (v) *Separability:* $(\sum_{i=1}^n h_i(x_i))^*(v_1, \dots, v_n) = \sum_{i=1}^n h_i^*(v_i)$, where $(x_1, \dots, x_n) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_n$.
- (vi) *Conjugate of a sum:* assume $h_1, h_2 \in \Gamma_0(\mathcal{H})$ and the relative interiors of their domains have a nonempty intersection. Then

$$(h_1 + h_2)^* = h_1^* \overset{\dagger}{\vee} h_2^* .$$

- (vii) For $V \in \mathbb{S}_{++}(N)$, the conjugate of f in \mathcal{H}_V is $h^*(Vu)$.

LEMMA A.5 (Conjugate of a degenerate quadratic function). *Let Q be a symmetric positive semi-definite matrix. Let Q^+ be its Moore-Penrose pseudo-inverse. Then,*

$$\left(\frac{1}{2} \|y - \cdot\|_Q^2 \right)^*(v) = \begin{cases} \frac{1}{2} \|y - v\|_{Q^+}^2 & \text{if } v \in y + \text{Im}(Q) , \\ +\infty & \text{otherwise .} \end{cases}$$

LEMMA A.6 (Conjugate of a rank-1 quadratic function). *Let $u \in \mathcal{H}$. Then,*

$$\left(\frac{1}{2} \langle u, \cdot \rangle^2 \right)^*(v) = \begin{cases} \frac{\|v\|^2}{2\|u\|^2} & \text{if } v \in \mathbb{R}u , \\ +\infty & \text{otherwise.} \end{cases}$$

Subdifferential.

DEFINITION A.7 (Subdifferential). *The subdifferential of a proper convex function $h \in \Gamma_0(\mathcal{H})$ at $x \in \mathcal{H}$ is the set-valued map $\partial h : \mathcal{H} \rightarrow 2^{\mathcal{H}}$*

$$\partial h(x) = \{v \in \mathcal{H} \mid \forall z \in \mathcal{H}, h(z) \geq h(x) + \langle v, z - x \rangle\} .$$

An element v of ∂h is called a subgradient. The subdifferential map ∂h is a maximal monotone operator from $\mathcal{H} \rightarrow 2^{\mathcal{H}}$.

LEMMA A.8. *If h is (Gâteaux) differentiable at x , its only subgradient at x is its gradient $\nabla h(x)$.*

LEMMA A.9. *Let $V \in \mathbb{S}_{++}(N)$. Then $V\partial h$ is the subdifferential of h in \mathcal{H}_V .*

The duality formulae to be stated shortly will be very useful throughout the rest of the paper.

Fenchel duality.

LEMMA A.10. *Let $h \in \Gamma_0(\mathcal{H})$ and $g \in \Gamma_0(\mathcal{H})$. Suppose that $0 \in \text{ri}(\text{dom } g - \text{dom } h)$. Then*

$$\inf_{x \in \mathcal{H}} h(x) + g(x) = - \min_{u \in \mathcal{H}} h^*(-u) + g^*(u) , \quad (\text{A.1})$$

with the extremality relationships between x^* and u^* , respectively the solutions of the primal and dual problems

$$\begin{aligned} x^* \in \partial h^*(-u^*) & \quad \text{and} \quad u^* \in \partial g(x^*) , \\ -u^* \in \partial h(x^*) & \quad \text{and} \quad x^* \in \partial g^*(u^*) . \end{aligned} \quad (\text{A.2})$$

Toland duality.

LEMMA A.11. *Let $h \in \Gamma_0(\mathcal{H})$ and $g \in \Gamma_0(\mathcal{H})$. Then*

$$\inf_{x \in \mathcal{H}} h(x) - g(x) = \min_{u \in \mathcal{H}} g^*(u) - h^*(u) . \quad (\text{A.3})$$

If $h - g$ is coercive, and u^* solves the dual problem in u , then there exists a solution x^* of the primal problem and

$$\begin{aligned} x^* \in \partial h^*(u^*) & \quad \text{and} \quad u^* \in \partial g(x^*) , \\ u^* \in \partial h(x^*) & \quad \text{and} \quad x^* \in \partial g^*(u^*) . \end{aligned} \quad (\text{A.4})$$

Proof. The first assertion is a consequence of [61, Theorem 2.2]. The extremality relationships follow by combining [61, Theorem 2.7 and 2.8].

□

A.2. Proximal calculus in \mathcal{H} .

DEFINITION A.12 (Moreau envelope [48]). *The function $\rho h(x) = \inf_{z \in \mathcal{H}} \frac{1}{2\rho} \|x - z\|^2 + h(z)$ for $0 < \rho < +\infty$ is the Moreau envelope of index ρ of h .*

ρh is also the infimal convolution of h with $\frac{1}{2\rho} \|\cdot\|^2$.

LEMMA A.13.

(i) *Translation:* $\text{prox}_{h(\cdot - y)}(x) = y + \text{prox}_h(x - y)$.

(ii) *Scaling:* $\forall \rho \in (-\infty, \infty), \text{prox}_{h(\rho \cdot)}(x) = \text{prox}_{\rho^2 f}(\rho x) / \rho$.

(iii) *Separability:* let $(h_i)_{1 \leq i \leq n}$ a family of functions each in $\Gamma_0(\mathbb{R})$ and $h(x) = \sum_{i=1}^N h_i(x_i)$.

Then h is in $\Gamma_0(\mathcal{H})$ and $\text{prox}_h = (\text{prox}_{h_i})_{1 \leq i \leq N}$.

LEMMA A.14. Let $h \in \Gamma_0(\mathcal{H})$. Then its Moreau envelope ${}^\rho h$ is convex and Fréchet-differentiable with $1/\rho$ -Lipschitz gradient

$$\nabla {}^\rho h = (\text{Id} - \text{prox}_{\rho h})/\rho.$$

LEMMA A.15 (Moreau identity). Let $h \in \Gamma_0(\mathcal{H})$, then for any $x \in \mathcal{H}$

$$\text{prox}_{\rho h^*}(x) + \rho \text{prox}_{h/\rho}(x/\rho) = x, \forall 0 < \rho < +\infty.$$

From Lemma A.15, we conclude that

$$\text{prox}_{h^*} = \text{Id} - \text{prox}_h, \quad \text{prox}_{h^*}(x) \in \partial h(x).$$

Appendix B. Proofs of Section 3.

B.1. Proof of Lemma 3.2. *Proof.* Let $p = \text{prox}_h^V(x)$. The statement follows from the following equivalences

$$\begin{aligned} p = \text{prox}_h^V(x) &\Leftrightarrow x \in p + V^{-1}\partial h(p) \\ &\Leftrightarrow V^{1/2}x \in V^{1/2}p + V^{-1/2} \circ \partial h \circ V^{-1/2}(V^{1/2}p) \\ &\Leftrightarrow V^{1/2}p = \text{prox}_{h \circ V^{-1/2}}(V^{1/2}x). \end{aligned}$$

□

B.2. Proof of Lemma 3.3. *Proof.* We have

$$\begin{aligned} p = \text{prox}_{\rho h^*}^V(x) &= (\text{Id} + V^{-1}\rho\partial h^*)^{-1}(x) \Leftrightarrow V(x - p) \in \partial(\rho h^*)(p) \\ &\Leftrightarrow p \in \partial h(V(x - p)/\rho) \\ &\Leftrightarrow Vx/\rho - (Vx - Vp)/\rho \in V\partial(h/\rho)(V(x - p)/\rho) \\ &\Leftrightarrow V(x - p)/\rho = (\text{Id} + V\partial(h/\rho))^{-1}(Vx) \\ &\Leftrightarrow x = p + \rho V^{-1} \circ (\text{Id} + V\partial(h/\rho))^{-1}(Vx). \end{aligned}$$

□

B.3. Proof of Theorem 3.4. *Proof.* Let $p = \text{prox}_h^V(x)$. Then, we have to solve

$$\begin{aligned}
& \min_z \frac{1}{2} \|x - z\|_V^2 + h(z) \\
& \Leftrightarrow \min_z \left(\frac{1}{2} \|z\|_P^2 - \langle x, z \rangle_P + h(z) \right) \pm \frac{1}{2} \langle x - z, Q(x - z) \rangle \\
\left[\begin{array}{l} y = P^{1/2}z \\ W = P^{-1/2}QP^{-1/2} \end{array} \right] & \Leftrightarrow \min_y \left(\frac{1}{2} \|y\|^2 - \langle P^{1/2}x, y \rangle + h \circ P^{-1/2}(y) \right) \pm \frac{1}{2} \langle P^{1/2}x - y, W(P^{1/2}x - y) \rangle \\
& \tag{B.1} \\
\left[\begin{array}{l} \text{Lemma A.10(A.1)} \\ \text{or Lemma A.11(A.3)} \end{array} \right] & \Leftrightarrow \min_w \pm \left(\frac{1}{2} \|\cdot\|^2 - \langle P^{1/2}x, \cdot \rangle + h \circ P^{-1/2} \right)^* (\mp w) + \left(\frac{1}{2} \langle P^{1/2}x - \cdot, W(P^{1/2}x - \cdot) \rangle \right)^* (w) \\
\left[\begin{array}{l} \text{Lemma A.5} \\ \text{and Lemma A.4(iv)} \end{array} \right] & \Leftrightarrow \min_{w \in \text{Im}(W)} \pm \left(\frac{1}{2} \|\cdot\|^2 - \langle P^{1/2}x, \cdot \rangle + h \circ P^{-1/2} \right)^* (\mp w) + \frac{1}{2} \|w\|_{W^+}^2 + \langle P^{1/2}x, w \rangle \\
\left[\text{Lemma A.4(vi)-(iii)} \right] & \Leftrightarrow \min_{w \in \text{Im}(W)} \pm \left(\left(\frac{1}{2} \|\cdot\|^2 - \langle P^{1/2}x, \cdot \rangle \right)^* \overset{+}{\vee} (h^* \circ P^{1/2}) \right) (\mp w) + \frac{1}{2} \|w\|_{W^+}^2 + \langle P^{1/2}x, w \rangle \\
& \Leftrightarrow \min_{w \in \text{Im}(W)} \pm \left(\left(\frac{1}{2} \|P^{1/2}x + \cdot\|^2 \right)^* \overset{+}{\vee} (h^* \circ P^{1/2}) \right) (\mp w) + \frac{1}{2} \|w\|_{W^+}^2 + \langle P^{1/2}x, w \rangle \\
\left[\text{Definition A.12} \right] & \Leftrightarrow \min_{w \in \text{Im}(W)} \pm^1 (h^* \circ P^{1/2})(P^{1/2}x \mp w) + \frac{1}{2} \|w\|_{W^+}^2 + \langle P^{1/2}x, w \rangle . \\
& \tag{B.2}
\end{aligned}$$

By virtue of Lemma A.14, ${}^1(h^* \circ P^{1/2})$ is continuously differentiable with 1-Lipschitz gradient. Together with Lemma A.8, Lemma A.10(A.2) or Lemma A.11(A.4)¹, and Lemma A.15, this yields

$$\begin{aligned}
p &= P^{-1/2} \circ \nabla^1 (h^* \circ P^{1/2})(P^{1/2}x \mp w^*) = P^{-1/2} \circ (\text{Id} - \text{prox}_{h^* \circ P^{1/2}})(P^{1/2}x \mp w^*) \\
&= P^{-1/2} \circ \text{prox}_{h \circ P^{-1/2}} \circ P^{1/2}(x \mp P^{-1/2}w^*),
\end{aligned}$$

where w^* is a solution to the dual problem (B.2), which will turn out to be unique as we will show shortly. Problem (B.2) is a minimization problem of a proper continuously differentiable objective with a Lipschitz continuous gradient over a linear set. The linear set can be parametrized by $\alpha \in \mathbb{R}^r$ such that $w = P^{-1/2}U\alpha$, and minimizing (B.2) is then equivalent to solving the r -dimensional smooth optimization problem

$$\min_{\alpha \in \mathbb{R}^r} \pm^1 (h^* \circ P^{1/2})(P^{1/2}x \mp P^{-1/2}U\alpha) + \frac{1}{2} \|\alpha\|_{U^\top Q^+ U}^2 + \langle U^\top x, \alpha \rangle . \tag{B.3}$$

The gradient of the objective in (B.3) is given by the function $l(\alpha)$. Lipschitz continuity of $l(\alpha)$ follows from non-expansiveness of the proximal mapping, and the Lipschitz constant is straightforward from the triangle and Cauchy–Schwartz inequality. The root α^* of l is unique, since l is strongly monotone. In the case $V = P + Q$, this is directly clear, since all terms in (B.3) are convex, and $\|\alpha\|_{U^\top Q^+ U}^2$ is strongly convex, since $X = U^\top Q^+ U^\top$ is positive definite. In case $V = P - Q$, we apply Moreau’s identity $(-{}^1(\varphi^*))(x) = {}^1\varphi(x) - \frac{1}{2} \|x\|^2$ for $\varphi \in \Gamma_0(\mathcal{H})$ (see, for example, [62, Lemma 2.10]) to the first term, which reduces the analysis of strong convexity to that of $\langle \alpha, (U^\top(Q^{-1} - P^{-1})U)\alpha \rangle$, hence, the positive definiteness of

¹The coercivity assumption holds (in fact the primal has exactly one solution) and the dual problem has indeed a non-empty set of minimizers.

$U^\top(Q^{-1} - P^{-1})U$. Since $P - Q \in \mathbb{S}_{++}(N)$, we have $\|P^{-1/2}QP^{-1/2}\| < 1$ and $P^{-1/2}QP^{-1/2}$ is invertible on $\text{Im}(Q)$. Therefore, using $1 = \|AA^{-1}\| \leq \|A\| \|A^{-1}\|$ for an invertible matrix A , we conclude that $\|P^{1/2}Q^+P^{1/2}\|_{\text{Im}(Q)} > 1$, where $\|\cdot\|_{\text{Im}(Q)}$ denotes the operator norm restricted to $\text{Im}(Q)$, which implies that $Q^{-1} - P^{-1} \in \mathbb{S}_{++}(\text{Im}(Q))$ and, thus, (B.3) is strongly convex. \square

B.4. Proof of Proposition 3.7. *Proof.* This follows from linearity and the chain rule [50, Lemma 3.5]. The second statement is a consequence of strong monotonicity of l as established in Theorem 3.4. \square

B.5. Proof of Proposition 3.11. *Proof.* Recall that (3.10) is strictly increasing, continuous, and has a unique solution. When prox_{h_i/d_i} is piecewise affine with k_i segments, it is easy to see that $l(\alpha)$ in (3.10) is also piecewise affine with slopes and intercepts changing at the k' (unique) transition points $\tilde{\theta}$. Therefore, the root of l can be found by sorting $\tilde{\theta}$ (Step 1) and finding the interval between breakpoints that localizes the root (Step 3). Step 1 has the complexity $O(K \log(K))$. Step 3 has the complexity $O(N \log(K))$, where $O(\log(K))$ steps are required for binary search and each step costs the evaluation of l , which consists of N terms. Step 5 adds at most a complexity of $O(N)$. \square

Appendix C. Proofs of Section 4.

C.1. Proof of Lemma 4.1. *Proof.* From [63, p. 57 and 64], we have for any x and y in $\text{dom}(F)$

$$\begin{aligned} L^{-1} \|\nabla f(x) - \nabla f(y)\|^2 &\leq \langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L \|x - y\|^2 \\ \mu \|x - y\|^2 &\leq \langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \mu^{-1} \|\nabla f(x) - \nabla f(y)\|^2. \end{aligned}$$

Thus by applying the above results to the quasi-Newton sequences s_k and y_k , we get

$$\begin{aligned} L^{-1} \|y_k\|^2 \leq \langle s_k, y_k \rangle \leq L \|s_k\|^2 &\implies L^{-1} \leq \frac{\langle s_k, y_k \rangle}{\|y_k\|^2} \leq \mu^{-1} \\ \mu \|s_k\|^2 \leq \langle s_k, y_k \rangle \leq \mu^{-1} \|y_k\|^2 &\implies \mu \leq \frac{\langle s_k, y_k \rangle}{\|s_k\|^2} \leq L. \end{aligned} \quad (\text{C.1})$$

We will use the ‘‘2nd’’ Barzilai–Borwein stepsize τ_{BB2} as opposed to the more common τ_{BB1} :

$$\tau_{\text{BB2}} = \frac{\langle s_k, y_k \rangle}{\|y_k\|^2}, \quad \tau_{\text{BB1}} = \frac{\|s_k\|^2}{\langle s_k, y_k \rangle}.$$

Via Cauchy-Schwarz, we have $\tau_{\text{BB2}} \leq \tau_{\text{BB1}}$. From (C.1), we have $L^{-1} \leq \tau_{\text{BB2}} \leq \tau_{\text{BB1}} \leq \mu^{-1}$.

Given the SR1 update and the choice $H_0 = \gamma \tau_{\text{BB2}} \text{Id}$ with $0 < \gamma < 1$, we have

$$u_k = (s_k - \gamma \tau_{\text{BB2}} y_k) / \sqrt{\langle s_k - \gamma \tau_{\text{BB2}} y_k, y_k \rangle} = (s_k - \gamma \tau_{\text{BB2}} y_k) / \sqrt{(1 - \gamma) \langle s_k, y_k \rangle}.$$

Combining this with the estimates (C.1), we obtain

$$\begin{aligned} \|u_k\|^2 &= \frac{\|s_k\|^2 - 2\gamma \tau_{\text{BB2}} \langle s_k, y_k \rangle + \gamma^2 \tau_{\text{BB2}}^2 \|y_k\|^2}{(1 - \gamma) \langle s_k, y_k \rangle} \\ &= (1 - \gamma)^{-1} \left(\frac{\|s_k\|^2}{\langle s_k, y_k \rangle} - 2\gamma \tau_{\text{BB2}} + \gamma^2 \tau_{\text{BB2}} \right) \\ &\leq (1 - \gamma)^{-1} (\mu^{-1} - 2\gamma L^{-1} + \gamma^2 \mu^{-1}). \end{aligned}$$

Thus

$$\begin{aligned} 0 &< \gamma L^{-1} \text{Id} \preceq H_0 \preceq H_k \preceq \gamma \mu^{-1} \text{Id} + (1 - \gamma)^{-1} \left((1 + \gamma^2) \mu^{-1} - 2\gamma L^{-1} \right) \text{Id} \\ &\preceq (1 - \gamma)^{-1} \left((1 + \gamma) \mu^{-1} - 2\gamma L^{-1} \right) \text{Id}. \end{aligned}$$

□

C.2. Proof of Theorem 4.2. *Proof.* We first recall the classical inequality for smooth functions with L -Lipschitz continuous gradient,

$$f(x) - f(y) + \langle \nabla f(y), y - x \rangle \leq \frac{L}{2} \|x - y\|^2. \quad (\text{C.2})$$

- *Case* $\alpha \in]0, 1/2[$: It is clear that (2.3) is equivalent to

$$B_k(x_k - x_{k+1}) - \kappa_k \nabla f(x_k) \in \kappa_k \partial h(x_k)$$

which in turn implies

$$h(y) \geq h(x_{k+1}) + \kappa_k^{-1} \langle B_k(x_k - x_{k+1}) - \kappa_k \nabla f(x_k), y - x_{k+1} \rangle, \quad \forall y \in \text{dom}(h). \quad (\text{C.3})$$

Applied at x_k , it yields

$$h(x_k) - h(x_{k+1}) + \langle \nabla f(x_k), x_k - x_{k+1} \rangle \geq \kappa_k^{-1} \|x_{k+1} - x_k\|_{B_k}^2. \quad (\text{C.4})$$

Denote $D_k = h(x_k) - h(x_{k+1}) + \langle \nabla f(x_k), x_k - x_{k+1} \rangle$. We have $D_k \geq 0$. In view of (C.2), we get

$$\begin{aligned} F(x_{k+1}) - F(x_k) + D_k &= f(x_{k+1}) - f(x_k) + \langle \nabla f(x_k), x_k - x_{k+1} \rangle \\ &\leq \frac{L}{2} \|x_{k+1} - x_k\|^2 \leq \frac{Lb}{2} \|x_{k+1} - x_k\|_{B_k}^2, \end{aligned}$$

where we used Lemma 4.1. The last inequality together with (C.4) yields

$$F(x_{k+1}) - F(x_k) \leq - \left(1 - \frac{Lb\kappa_k}{2} \right) D_k \leq -\alpha D_k.$$

By assumption, the right hand side is non-positive, meaning that the objective function decreases with k . Denote

$$E_k = F(x_k) - F(x^*) \quad \text{and} \quad \Delta_k = E_k - E_{k+1}.$$

Observe that E_k is a positive and decreasing sequence, and thus converges. Moreover,

$$\Delta_k \geq \alpha D_k,$$

Using convexity of f and inequality (C.3) at $y = x^*$, we obtain

$$\begin{aligned} E_k &= f(x_k) - f(x^*) + \langle \nabla f(x_k), x^* - x_k \rangle \\ &\quad + h(x_k) - h(x_{k+1}) + \langle \nabla f(x_k), x_k - x_{k+1} \rangle \\ &\quad + h(x_{k+1}) - h(x^*) + \langle \nabla f(x_k), x_{k+1} - x^* \rangle \\ &\leq D_k + \kappa_k^{-1} \langle B_k(x_k - x_{k+1}), x_{k+1} - x^* \rangle \\ &\leq D_k + \kappa_k^{-1} \|x_{k+1} - x^*\|_{B_k} \|x_{k+1} - x_k\|_{B_k} \\ &\leq D_k + \sqrt{\frac{1}{\kappa a}} \|x_{k+1} - x^*\| \sqrt{D_k} \\ &\leq \alpha^{-1} \left(\Delta_k + \sqrt{\frac{\alpha}{\kappa a}} \|x_{k+1} - x^*\| \sqrt{\Delta_k} \right). \end{aligned}$$

Thus, using Young inequality, together with strong convexity of f and E_k is decreasing, we get for any $\varepsilon > 0$,

$$\begin{aligned}
\alpha E_k &\leq \Delta_k + \frac{\alpha\varepsilon}{2\kappa a} \|x_{k+1} - x^*\|^2 + \frac{\Delta_k}{2\varepsilon} \\
&\leq \left(1 + \frac{1}{2\varepsilon}\right) \Delta_k + \frac{\alpha\varepsilon}{\kappa a \mu} E_{k+1} \\
&\leq \left(1 + \frac{1}{2\varepsilon}\right) \Delta_k + \frac{\alpha\varepsilon}{\kappa a \mu} E_k \\
&= \left(1 + \frac{1}{2\varepsilon}\right) \Delta_k + \frac{\varepsilon L \alpha}{\gamma \kappa \mu} E_k .
\end{aligned}$$

Let $\beta(\varepsilon) = 1 + \frac{1}{2\varepsilon}$ and $c = L/\mu > 1$. It follows that

$$E_{k+1} \leq \rho E_k , \quad \rho = 1 - \frac{\alpha}{\beta(\varepsilon)} \left(1 - \frac{\varepsilon c}{\gamma \kappa}\right) .$$

We always have $\beta(\varepsilon) \in]1, +\infty[$, and by assumption on the sequence κ_k , $\alpha \in]0, 1[$. Choosing $\varepsilon = \nu \gamma \kappa / c$, for any $\nu \in]0, 1[$, we get that $\rho = 1 - \alpha \frac{\nu(1-\nu)}{\nu+\eta} \in]0, 1[$. Therefore,

$$\|x_k - x^*\| \leq \sqrt{\frac{2(F(x_0) - (Fx^*))}{\mu}} \rho^{k/2} .$$

The function $\nu \in]0, 1] \mapsto \nu(1-\nu)/(\nu+\eta)$ has a unique maximizer at $\nu_{\text{opt}} = \sqrt{\eta^2 + \eta} - \eta$ (which is indeed a strictly increasing function of η on $]0, +\infty[$ taking values in $]0, 1/2[$). We get the optimal rate ρ_1 by plugging ν_{opt} into the expression of ρ .

- *Case* $\alpha \in [1/2, 1[$: From (2.1), (C.2), Lemma 4.1 and the assumption on α , we have

$$\begin{aligned}
Q_k^{B_k}(x) + h(x) &= F(x) - (f(x) - f(x_k) + \langle \nabla f(x_k), x_k - x \rangle) + \frac{1}{2\kappa_k} \|x - x_k\|_{B_k}^2 \\
&\geq F(x) - \frac{L}{2} \|x - x_k\|^2 + \frac{1}{2\kappa_k} \|x - x_k\|_{B_k}^2 \\
&\geq F(x) + \frac{1}{2\kappa_k} (1 - Lb\kappa_k) \|x - x_k\|_{B_k}^2 \\
&\geq F(x) .
\end{aligned}$$

Moreover, convexity of f yields

$$\begin{aligned}
Q_k^{B_k}(x_{k+1}) + h(x_{k+1}) &= \min_x Q_k^{B_k}(x) + h(x) \\
&= \min_x F(x) - (f(x) - f(x_k) - \langle \nabla f(x_k), x - x_k \rangle) + \frac{1}{2\kappa_k} \|x - x_k\|_{B_k}^2 \\
&\leq \min_x F(x) + \frac{1}{2\kappa_k} \|x - x_k\|_{B_k}^2 .
\end{aligned}$$

It then follows that

$$\begin{aligned}
F(x_{k+1}) &\leq Q_k^{B_k}(x_{k+1}) + h(x_{k+1}) \\
&\leq \min_x F(x) + \frac{1}{2\kappa_k} \|x - x_k\|_{B_k}^2 \\
&\leq \min_{t \in [0,1]} F(tx^* + (1-t)x_k) + \frac{t^2}{2\kappa_k} \|x_k - x^*\|_{B_k}^2 \\
&\leq \min_{t \in [0,1]} tF(x^*) + (1-t)F(x_k) + \frac{t^2 L}{2\kappa\gamma} \|x_k - x^*\|^2 \\
&\leq \min_{t \in [0,1]} F(x_k) - t(F(x_k) - F(x^*)) + \frac{t^2 L}{\kappa\gamma\mu} (F(x_k) - F(x^*)) \\
&= \min_{t \in [0,1]} F(x_k) - t(1-2t\eta) (F(x_k) - F(x^*)) .
\end{aligned}$$

Thus, we arrive at

$$E_{k+1} \leq \min_{t \in [0,1]} (1 - t(1 - 2t\eta)) E_k = \rho_2 E_k .$$

The function $(1 - t(1 - 2t\eta))$ attains its minimum uniquely at $t = 1$ if $\eta \leq 1/4$, and $1/(4\eta)$ otherwise. Plugging these values gives the expression of ρ_2 . \square

Appendix D. Proofs of Section 5.

D.1. Proof of Lemma 5.1. *Proof.* We derive a uniform bound for the matrix H in (5.1). Note that u_γ from (5.1) satisfies with $\tau = \tau_{\text{BB}2}$

$$\begin{aligned}
\rho \|u_\gamma\|^2 &= \frac{\|s\|^2 - 2\frac{\gamma\tau}{1+\gamma} \langle y, s \rangle + \left(\frac{\gamma\tau}{1+\gamma}\right)^2 \|y\|^2}{\langle y, s \rangle} = \frac{\|s\|^2}{\langle y, s \rangle} - 2\frac{\gamma}{1+\gamma} \tau_{\text{BB}2} + \frac{\gamma^2}{(1+\gamma)^2} \tau_{\text{BB}2} \\
&= \tau_{\text{BB}1} - \frac{\gamma}{1+\gamma} \left(2 - \frac{\gamma}{1+\gamma}\right) \tau_{\text{BB}2} \leq \mu^{-1} - \frac{(2+\gamma)\gamma}{(1+\gamma)^2} L^{-1}
\end{aligned}$$

where we used $\rho^{-1} = \langle y, s \rangle$, $\rho \|y\|^2 = \tau_{\text{BB}2}^{-1}$, and the estimations in the proof of Lemma 4.1 for $\tau_{\text{BB}2}$ and $\tau_{\text{BB}1}$. Using this estimation, $\rho\tau_{\text{BB}2}^2 \|y\|^2 = \tau_{\text{BB}2} \geq L^{-1}$, and positive semi-definiteness of yy^\top and $u_\gamma u_\gamma^\top$, we conclude that

$$\begin{aligned}
0 \prec \frac{\gamma}{1+\gamma} L^{-1} \text{Id} &= \gamma \left(\tau_{\text{BB}2} - \frac{\gamma^2}{1+\gamma} \tau_{\text{BB}2} \right) \text{Id} \preceq H \preceq \tau_{\text{BB}2} \gamma \text{Id} + (1+\gamma) \mu^{-1} \text{Id} - \frac{(2+\gamma)\gamma}{1+\gamma} L^{-1} \text{Id} \\
&\preceq (1+2\gamma) \mu^{-1} \text{Id} - \frac{(2+\gamma)\gamma}{1+\gamma} L^{-1} \text{Id}.
\end{aligned}$$

\square

REFERENCES

- [1] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer-Verlag, New York, 2011.
- [2] S. Becker and J. Fadili. A quasi-Newton proximal splitting method. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2618–2626. Curran Associates Inc., 2012.
- [3] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Computing*, 16(5):1190–1208, 1995.

- [4] P.L. Combettes and B.C. Vũ. Variable metric forward–backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289–1318, September 2014.
- [5] B. C. Vũ. A variable metric extension of the Forward–Backward–Forward algorithm for monotone operators. *Numerical Functional Analysis and Optimization*, 34(9):1050–1065, 2013.
- [6] S. Karimi and S. Vavasis. IMRO: A proximal quasi-newton method for solving ℓ_1 -regularized least squares problems. *SIAM Journal on Optimization*, 27(2):583–615, January 2017.
- [7] P. L. Combettes and J. C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, New York, 2011.
- [8] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.
- [9] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57, 2009. 2479–2493.
- [10] J. Barzilai and J. Borwein. Two point step size gradient method. *IMA J. Numer. Anal.*, 8:141–148, 1988.
- [11] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. on Imaging Sci.*, 2(1):183–202, 2009.
- [12] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.
- [13] W. Su, S. Boyd, and R. Candes. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.
- [14] J. Attouch, H; Peypouquet. The rate of convergence of nesterov’s accelerated forward-backward method is actually faster than $1/k^2$. *SIAM Journal on Optimization*, 26(3):1824–1834, 2016.
- [15] Brendan B. O’Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, June 2015.
- [16] P. Ochs and T. Pock. Adaptive Fista. *ArXiv e-prints*, November 2017. arXiv: 1711.04343 [math].
- [17] G. HG Chen and R. T. Rockafellar. Convergence rates in Forward–Backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, 1997.
- [18] P.L. Combettes and B.C. Vũ. Variable metric quasi-Fejér monotonicity. *Nonlinear Analysis: Theory, Methods & Applications*, 78:17–31, February 2013.
- [19] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *International Conference on Computer Vision (ICCV)*, 2011.
- [20] K. Bredies and H. Sun. Preconditioned Douglas–Rachford splitting methods for convex-concave saddle-point problems. *SIAM Journal on Numerical Analysis*, 53(1):421–444, 2015.
- [21] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.
- [22] José Luis Morales and Jorge Nocedal. Remark on algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization: *ACM Transactions on Mathematical Software*, 38(1):7:1–7:4, 2011.
- [23] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, 17:526–557, 2006.
- [24] G. Andrew and J. Gao. Scalable training of l_1 -regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning, ICML’07*, pages 33–40, New York, NY, USA, 2007. ACM.
- [25] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l_1 regularization: A comparative study and two new approaches. In *European Conference on Machine Learning*, 2007.
- [26] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.
- [27] T. Goldstein and S. Setzer. High-order methods for basis pursuit. Technical report, CAM-UCLA, 2011.
- [28] J. Yu, S.V.N. Vishwanathan, S. Guenter, and N. Schraudolph. A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. *J. Machine Learning Research*, 11:1145–1200, 2010.
- [29] A.S. Lewis and M.L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, October 2013.

- [30] A.S. Lewis and S. Zhang. Nonsmoothness and a variable metric method. *Journal of Optimization Theory and Applications*, 165(1):151–171, April 2015.
- [31] J. Guo and A. Lewis. BFGS convergence to nonsmooth minimizers of convex functions. *ArXiv e-prints*, March 2017. arXiv: 1703.06690.
- [32] C. Lemaréchal. Numerical experiments in nonsmooth optimization. In E.A. Nurminski, editor, *Progress in Nondifferentiable Optimization*, IIASA, pages 61–84. Laxenburg, 1982.
- [33] M. Schmidt, E. van den Berg, M. Friedlander, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *AISTATS*, 2009.
- [34] M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.
- [35] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25(1), 2009.
- [36] S. Bonettini and M. Prato. New convergence results for the scaled gradient projection method. *Inverse Problems*, 31(9), 2015.
- [37] M. P. Friedlander and G. Goh. Efficient evaluation of scaled proximal operators. *Electronic Transactions on Numerical Analysis*, 46:1–22, 2017.
- [38] J. Lee, Y. Sun, and M. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, January 2014.
- [39] S. Bonettini, I. Loris, F. Porta, and M. Prato. Variable metric inexact line-search based methods for nonsmooth optimization. *SIAM Journal on Optimization*, 26(2):891–921, January 2016.
- [40] S. Salzo. The variable metric forward–backward splitting algorithm under mild differentiability assumptions. *ArXiv e-prints*, May 2016. arXiv:1605.00952 [math].
- [41] Roger R. Fletcher. A limited memory steepest descent method. *Mathematical Programming*, 135(1–2):413–436, July 2011.
- [42] F. Porta, M. Prato, and L. Zanni. A new steplength selection for scaled gradient methods with application to image deblurring. *Journal of Scientific Computing*, 65(3):895–919, December 2015.
- [43] S. Bonettini, I. Loris, F. Porta, M. Prato, and S. Rebegoldi. On the convergence of variable metric line-search based proximal-gradient method under the Kurdyka-Lojasiewicz inequality. *ArXiv e-prints*, May 2016. arXiv:1605.03791 [math].
- [44] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, November 2013.
- [45] P. Patrinos, L. Stella, and A. Bemporad. Forward–backward truncated Newton methods for convex composite optimization. *ArXiv e-prints*, February 2014. arXiv:1402.6655 [math].
- [46] L. Stella, A. Themelis, and P. Patrinos. Forward–backward quasi-Newton methods for nonsmooth optimization problems. *Computational Optimization and Applications*, 67(3):443–487, July 2017.
- [47] P. Ochs. Unifying abstract inexact convergence theorems for descent methods and block coordinate variable metric iPiano. *ArXiv e-prints*, February 2016. arXiv:1602.07283.
- [48] J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CRAS Séries A Mathematics*, 255:2897–2899, 1962.
- [49] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems I and II*. Springer, New York, 2003.
- [50] R. Griesse and D. A. Lorenz. A semismooth Newton method for Tikhonov functionals with sparsity constraints. *Inverse Problems*, 24(3):035007, 2008.
- [51] C. Broyden. Quasi-Newton methods and their application to function minimization. *Mathematics of Computation*, 21:577–593, 1967.
- [52] N. Gould. Seminal papers in nonlinear optimization. In *An introduction to algorithms for continuous optimization*. Oxford University Computing Laboratory, 2006. <http://www.numerical.rl.ac.uk/nimg/course/lectures/paper/paper.pdf>.
- [53] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [54] I. Dhillon, D. Kim, and S. Sra. Tackling box-constrained optimization via a new projected quasi-Newton approach. *SIAM Journal on Scientific Computing*, 32(6):3548–3563, 2010.
- [55] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.
- [56] M. Schmidt. *Graphical Model Structure Learning with L1-Regularization*. PhD thesis, University of British Columbia, Vancouver, 2010. <https://www.cs.ubc.ca/~schmidtm/Software/thesis.html>.

- [57] Roger Fletcher. On the Barzilai-Borwein method. In Liqun Qi, Koklay Teo, Xiaoqi Yang, Panos M. Pardalos, and Donald W. Hearn, editors, *Optimization and Control with Applications*, volume 96 of *Applied Optimization*, pages 235–256. Springer US, 2005.
- [58] H. Raguet, J. Fadili, and G. Peyré. A generalized forward–backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226, January 2013.
- [59] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing*, 62:89–108, 1999.
- [60] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
- [61] J.F. Toland. A duality principle for non-convex optimisation and the calculus of variations. *Archive for Rational Mechanics and Analysis*, 71(1):41–61, 1979.
- [62] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward–backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.
- [63] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer, Boston, 2004.