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 piecewise 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 the 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

DOI. 10.1137/18M1167152

1. Introduction. Convex optimization has proved to be extremely useful in 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 nonlinear 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 nonsmooth 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 nonsmooth and "simple" in a certain way. The gradient descent method has a natural extension to these structured nonsmooth optimization problems, which is known as *proximal gradient descent* (which includes projected gradient descent as a subcase) or *forward-backward splitting* [5]. Algorithmically, besides a gradient step with respect to the smooth term of the objective, the generalization requires the solution of proximal subproblems with respect to the nonsmooth term of the objective. The "simple" property from above refers to the proximal subproblems. In many situations, these subproblems can be solved analytically or very efficiently. However, a

[†]Department of applied mathematics, University of Colorado, 526 UCB, 80309 Boulder, CO (stephen.becker@colorado.edu).

[‡]ENSICAEN, 6 Bd. Maréchal Juin, 14050 Caen, France (jalal.fadili@ensicaen.fr).

[§]Faculty of Mathematics and Computer Science, Saarland University, Campus E1.7, 66123 Saarbrücken, Germany (ochs@math.uni-sb.de).

change of the metric, which is the key feature of quasi-Newton methods or nonlinear 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 (symmetric rank-1) and the L-BFGS methods, to be efficiently applied to structured nonsmooth 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 nonsmooth optimization problems.

1.1. Problem statement. Let $\mathcal{H} = (\mathbb{R}^N, \langle \cdot, \cdot \rangle)$ be 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 (SPD) 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 the identity operator as Id. For a matrix A, A^+ is its Moore–Penrose pseudoinverse. For a positive semidefinite matrix A, $A^{1/2}$ denotes its principal square root.

An extended-valued function $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ is (0)-*coercive* if

$$\lim_{\|x\| \to +\infty} f(x) = +\infty.$$

The *domain* of f is defined by $\operatorname{dom} f = \{x \in \mathcal{H} \ : \ f(x) < +\infty\}$ and f is *proper* if $\operatorname{dom} f \neq \emptyset$. We say that a real-valued function f is *lower semicontinuous* (lsc) if $\liminf_{x \to x_0} f(x) \geq f(x_0)$. 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

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

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

The class we consider covers structured smooth + nonsmooth 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, $\lambda > 0$, and b be a vector of appropriate dimensions. Then the LASSO problem reads

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

Example 1.2 (nonnegative least-squares (NNLS)). Let A and b be as in Example 1.1. Then

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

Example 1.3 (sparse support vector machines). One would like to find a linear decision function which minimizes the objective

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

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

1.2. Contributions. We introduce a general proximal calculus in a metric $V = P \pm Q \in \mathbb{S}_{++}(N)$ given by $P \in \mathbb{S}_{++}(N)$ and a positive semidefinite rank-r matrix Q. This significantly extends the result in the preliminary version of this paper [7], 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 3.3.4 for a nonexhaustive 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) which 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. Whereas L-BFGS-B [16] is only applicable to box constraints, our quasi-Newton methods efficiently apply to a wide-variety of nonsmooth functions.

To clarify the differences between this paper and the conference paper [7], the current paper (1) extends the proximal framework to allow $V = P \pm Q$ scalings, where Q is rank $r \geqslant 1$ (Theorem 3.4, and specialized to the $r = 1$ case in Theorem 3.8), using Toland duality to handle nonconvexity issues that arise in the $P - Q$ case, whereas [7] considers only $V = P + Q$ for Q rank-1 and positive semidefinite; (2) discusses at length bisection and semismooth methods to solve the dual problem, and gives global (Proposition 3.11) and local (Proposition 3.7) convergence results, respectively; (3) introduces the zero-memory L-BFGS quasi-Newton forward-backward algorithm (Algorithm 3) in addition to the SR1 one; (4) proves convergence results for these algorithms (Theorems 4.2 and 5.2, respectively); and (5) discusses a few new examples of nonseparable proximity operators including that of the $\ell_1 - \ell_2$ norm in section 3.3.4 and runs numerical experiments with this norm in section 6.2.

1.3. Paper organization. Section 2 formally introduces quasi-Newton methods and their generalization to the structured nonsmooth 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 semismooth Newton-based 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:

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

where $B \in \mathbb{S}_{++}(N)$ and $\kappa > 0$. The (nonrelaxed) 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

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

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., [22, 73]) 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 sequence $(x_k)_{k \in \mathbb{N}}$ is expected when $0 < \underline{\kappa} \leqslant \kappa_k \leqslant \overline{\kappa} < 2/L$, where L is the Lipschitz constant of ∇f.

Note that when $h = 0$, (2.2) reduces to gradient descent if $B_k = \mathrm{Id}$, 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 (at least locally) if $\nabla^2 f(x^\star)$ is positive-definite, 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 $h = 0$ case, we pursue a quasi-Newton approximation for B_k for the $h \neq 0$ case. 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 restrict B_k to the structured form of a positive-definite "simple" (e.g., 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 (0SR1) and a "diagonal $-$ rank-2" metric (0BFGS) 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 the approximation to its inverse. 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.

In sections 4 and 5, we will prove Algorithm 1 converges linearly under the assumption that f is strongly convex and $t = 1$, which is the standard theoretically controllable setting for Newton and quasi-Newton methods. Moreover, global convergence of subsequences to a minimizer for the line-search variant can be deduced from

the literature [62, 10, 51]. Thanks to the line search, the choice of the metric *need not* obey monotonicity. If standard assumptions on the monotonicity of the metric are satisfied, convergence to a minimizer can be proved [62, 10]. Moreover, the convergence results in [10] account for inexact evaluation of the proximal mapping, which even allows us to invoke a semismooth Newton method for solving the subproblems numerically (see section 3.2.2).

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

Require: $x_0 \in \mathrm{dom}(f + h)$, stopping criterion ϵ, method to compute step sizes t and κ_k (e.g., based on the Lipschitz constant estimate L of ∇f and strong convexity μ of f)
 1: **for** $k = 1, 2, 3, \ldots$ **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 step size κ_k:

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

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

Algorithm 2 Zero-memory symmetric rank-1 (0SR1) algorithm to solve (P); cf. section 4.

Require: as for Algorithm 1, and parameters γ, $\tau_{\min}$, $\tau_{\max}$ for Algorithm 7
 Iterate as in Algorithm 1, with line 4 as
 4: Compute H_k via Algorithm 7 (diagonal plus rank-1)

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 (5.1) (diagonal plus rank-2)

Remark 2.1. The use of the terms "diagonal $-$ rank-r" and "diagonal $+$ rank-r" needs 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 [7] 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 [37].

2.2. Relation to prior work.

First-order methods. The algorithm in (2.2) with $B_k = \mathrm{Id}$ is variously known as the proximal gradient descent or iterated shrinkage/thresholding algorithm (IST or ISTA). It has a grounded convergence theory, and also admits over-relaxation factors $\alpha \in (0,1)$ [23].

The spectral projected gradient (SPG) [8] method was designed as an extension of the Barzilai–Borwein spectral step-length method to constrained problems. In [75], it was extended to nonsmooth problems by allowing general proximity operators. The Barzilai–Borwein method [4] 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 [6] is a (two-step) inertial version of ISTA inspired by the work of Nesterov [47]. 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 [68, 2]. While the step size 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 [3], since this led to improved performance.

Recently, in [52] it was shown that by optimizing the inertial parameter in each iteration of FISTA, applied to the sum of a quadratic function and a nonsmooth 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 [22, 73] 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 [21]. 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 so that the proximity operator is easily computable. When the metric is not diagonal but fixed, these methods can be viewed as preconditioned versions that were shown to perform well in practice for certain problems (i.e., functions h) [55, 14]. But again, the choice of the metric (preconditioner) 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 use a more advanced quadratic model to solve for the free variables. A well-known such method is L-BFGS-B [16, 77], which handles general box-constrained problems; we test an updated version [45]. A recent bound-constrained solver is ASA [36], which uses a CG solver on the free variables and shows good results compared to L-BFGS-B, SPG, GENCAN, and TRON. We also compare our algorithm to several active-set approaches specialized for ℓ_1 penalties: "orthant-wise Learning" (OWL) [1], "projected scaled sub-gradient + active set" (PSSas) [64], "fixed-point continuation + active set" (FPC_AS) [74], and "CG + IST" (CGIST) [32].

Other approaches. By transforming the problem into a standard conic programming problem, the generic problem is amenable to interior-point methods (IPMs).

IPMs require solving a Newton-step equation, so first-order-like "Hessian-free" variants of IPMs 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 nonsmooth h term in the quadratic approximation. Yu, Guenter, and Schraudolph [76] propose a nonsmooth 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; see [43, 44], and more recently [34]. The empirically justified good numerical performance has been observed for decades [42].

The projected quasi-Newton (PQN) algorithm [66, 65] is perhaps the most elegant and logical extension of quasi-Newton methods, but it involves solving a subiteration or needs to be restricted to a diagonal metric in the implementation [13, 12]. PQN implements 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 subiteration) is significantly more expensive to evaluate than projecting onto the constraints. Again, the cost of the subproblem solver (and a suitable stopping criteria for this inner solve) are issues. The paper [31] shows how the subproblem can be solved efficiently by a special interior-point method when h is a quadratic-support function. As discussed in [41], it is possible to generalize PQN to general nonsmooth 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, [10] proposes a flexible proximal quasi-Newton method that extends [12] 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 [62].

A more general and efficient step-size strategy with memory was proposed in [58] for unconstrained optimization, which was generalized to a scaled gradient projection method in [56], and used in the proximal gradient method in [11]. However, the flexible choice of the step size and the scaling of the metric is not for free when convergence guarantees are sought. Bonettini and coauthors [11, 10] rely on a line-search strategy to account for a descent of the objective values. The metric in [19] is constructed such that (2.1) is a majorizer of the (possibly nonconvex) objective and the step-size selection is more conservative; however, line search can be avoided.

The works [54, 67] 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 [50], and have been generalized further.

The proximal quasi-Newton methods described so far simply assume that the scaled proximal mapping can be solved efficiently, and rely on solving subproblems or simple diagonal scaling metrics. The first work on systematically solving nondiagonally scaled proximal mappings efficiently is the conference version of this paper [7]. The key is structure of the metric. In [7], 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 [37]. A systematic analysis for both cases where a rank-r modification is allowed is presented in this paper.

The key result for efficiently computing the proximal mapping in this paper reveals a decomposition into a simple proximal mapping (for example, w.r.t. a diagonal metric) and a low-dimensional operator equation (root-finding problem). In several cases, the operator equation can be solved exactly using specialized techniques. In the general case, we rely on a semismooth Newton strategy. It is known that the convergence of the latter, under mild conditions, is remarkably (locally) superlinear [28, 29], which may even be improved to quadratic convergence under strong semismoothness [57]. A similar result was independently obtained in [40] under similar assumptions.

Due to the great success of Newton's method for smooth equations, the nonsmooth setting has been actively studied and is still the subject of ongoing research; see for instance the recent monograph [71]. Early studies of generalizing Newton's method for solving nonsmooth equations include [38] for piecewise smooth equations, [53, 60] for so-called B-differentiable equations, and [39] for locally Lipschitz functions. As pointed out in [57], semismoothness is a crucial property in the superlinear convergence analysis of these methods. Semismooth Newton methods have also been adapted to nonsmooth operator equations in function spaces [70]. Recognizing semismoothness is, however, not always immediate. In [9], the authors proposed a large class of semismooth mappings. Our convergence results on the semismooth Newton method will then rely on [28, 29, 9].

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 to be 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 [46]). *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 $\mathrm{prox}_h(x)$. The single-valued operator $\mathrm{prox}_h : \mathcal{H} \to \mathcal{H}$ thus defined is the proximity operator or proximal mapping of h. Equivalently, $\mathrm{prox}_h = (\mathrm{Id} + \partial h)^{-1}$, where ∂h is the subdifferential of h. When h is the indicator function of a nonempty closed convex set $\mathcal{C}$, the corresponding proximity operator is the orthogonal projector onto $\mathcal{C}$, denoted $\mathrm{proj}_{\mathcal{C}}$.*

Throughout, we denote by

$$(3.1) \qquad \mathrm{prox}_h^V(x) = \operatorname*{argmin}_{z \in \mathcal{H}} h(z) + \frac{1}{2} \|x - z\|_V^2 = (\mathrm{Id} + V^{-1} \partial h)^{-1}(x)$$

the proximity operator of h w.r.t. the norm that $\mathcal{H}_V$ is endowed with 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:*

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

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 $x \in \mathcal{H}$,*

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

For $\rho = 1$, this simplifies to

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

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 $r \leqslant N$, given by r linearly independent vectors $u_1, \ldots, u_r \in \mathcal{H}$. 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 either exactly (see section 3.3) or by efficient fast iterative procedures with controlled complexity such as bisection (section 3.3.2) or semismooth Newton iterations (section 3.2.2).

3.2.1. General case. We start with our most general result.

THEOREM 3.4 (proximity operator for a rank-r modified metric). *Let $h \in \Gamma_0(\mathcal{H})$ and $V = P \pm Q \in \mathbb{S}_{++}(N)$, where $P \in \mathbb{S}_{++}(N)$ and $Q = \sum_{i=1}^r u_i u_i^\top \in \mathbb{R}^{N \times N}$ with $r = \text{rank}(Q) \leqslant N$. Let $U = (u_1, \ldots, u_r)$. Then,*

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

where $\alpha^\star \in \mathbb{R}^r$ is the unique zero of the mapping $\mathcal{L} : \mathbb{R}^r \to \mathbb{R}^r$,

$$\text{(3.5)} \qquad \begin{aligned} \mathcal{L}(\alpha) &:= U^\top \left(x - P^{-1/2} \circ \text{prox}_{h \circ P^{-1/2}} \circ P^{1/2}(x \mp P^{-1} U \alpha) \right) + \alpha \\ &= U^\top \left(x - \text{prox}_h^P(x \mp P^{-1} U \alpha) \right) + \alpha. \end{aligned}$$

The mapping $\mathcal{L}$ is Lipschitz continuous with Lipschitz constant $1 + \|P^{-1/2}U\|^2$, and strongly monotone with modulus c, where $c = 1$ for $V = P + Q$ and $c = 1 - \|P^{-1/2}U\|^2$ for $V = P - Q$.

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 dramatically reduces 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 (or equivalently $\text{prox}_{h \circ P^{-1/2}}$), which is often much easier to solve. This is typically the case when P is a diagonal matrix as will be considered in section 3.3. Another interesting scenario is when $h = \psi \circ P^{1/2}$, where $\psi \in \Gamma_0(\mathcal{H})$ is a simple function so that $\text{prox}_{h \circ P^{-1/2}} = \text{prox}_\psi$ is easy to compute. Thus the matrix P in the expression of V can be interpreted as a preconditioner. In section 3.3, we will focus on the case in which P is diagonal since all standard and efficient quasi-Newton methods (e.g., SR1, L-BFGS) use a diagonal P.

- 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 $\operatorname{rank}(Q) = r$, then

$$V^{-1} = P^{-1} \mp \widetilde{Q}^{-1}, \quad \widetilde{Q}^{-1} := P^{-1} Q \left(\operatorname{Id} \pm P^{-1} Q\right)^{-1} P^{-1},$$

with $\operatorname{rank}(\widetilde{Q}^{-1}) = r$. 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 (the Moreau identity in $\mathcal{H}_V$), the computation of the proximity operator of the convex conjugate function h^*, $\operatorname{prox}_{h^*}^V$, can be cast in terms of computing $\operatorname{prox}_h^{V^{-1}}$.

COROLLARY 3.6. *Let* $V = P + Q_1 - Q_2 \in \mathbb{S}_{++}(N)$ *with* $P \in \mathbb{S}_{++}(N)$ *and symmetric positive semidefinite matrices* Q_1, Q_2 *with* $\operatorname{rank}(Q_i) = r_i$ *and let* $\operatorname{Im}(Q_i)$ *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:*

$$\operatorname{prox}_h^V(x) = \operatorname{prox}_h^{P_1}(x + P_1^{-1} U_1 \alpha_1^\star) = \operatorname{prox}_h^P(x + P_1^{-1} U_1 \alpha_1^\star - P^{-1} U_2 \alpha_2^\star),$$

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

$$\mathcal{L}_1(\alpha_1, \alpha_2) = U_1^\top (x - \operatorname{prox}_h^P(x + P_1^{-1} U_1 \alpha_1 - P^{-1} U_2 \alpha_2)) - \alpha_1,$$
$$\mathcal{L}_2(\alpha_1, \alpha_2) = U_2^\top (x + P_1^{-1} U_1 \alpha_1 - \operatorname{prox}_h^P(x + P_1^{-1} U_1 \alpha_1 - P^{-1} U_2 \alpha_2)) - \alpha_2.$$

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 $P_1 = P + Q_1$. □

As discussed, depending on the structure of the proximity operator $\operatorname{prox}_{h \circ P^{-1/2}}$, either general-purpose or specialized algorithms for solving the root-finding problem can be derived. In some situations (see, e.g., Proposition 3.13) the root of the function $\mathcal{L}$ 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 we see now.

3.2.2. Semismooth Newton method. We here turn to the semismooth Newton method to solve $\mathcal{L}(\alpha) = 0$ (see (3.5)) using the fact that $\mathcal{L}$ is Lipschitz-continuous and strongly monotone (Theorem 3.4).

Since $\mathcal{L} : \mathbb{R}^r \to \mathbb{R}^r$ is Lipschitz continuous, it is so-called Newton differentiable [18], i.e., there exists a family of linear mappings $\mathcal{G}$ (called generalized Jacobians) such that for all α on an open subset of $\mathbb{R}^r$,

$$\lim_{d \to 0} \frac{\|\mathcal{L}(\alpha + d) - \mathcal{L}(\alpha) - \mathcal{G}(\alpha + d)d\|}{\|d\|} = 0.$$

However, this is only of little help algorithmically unless one can construct a generalized Jacobian $\mathcal{G}$ which is easily computable and provably invertible under our strong monotonicity assumption. This is why we turn to the semismoothness framework.

We shall write $J\mathcal{L}(\alpha) \in \mathbb{R}^{r \times r}$ for the usual Jacobian matrix whenever α is a point in the differentiability set $\Omega \subset \mathbb{R}^r$ (its complement has measure zero by the celebrated Rademacher theorem). The Clarke Jacobian of $\mathcal{L}$ at $\alpha \in \mathbb{R}^r$ is defined as [20, Definition 2.6.1]

$$\partial^C \mathcal{L}(\alpha) = \operatorname{conv} \left\{ G \in \mathbb{R}^{r \times r} : \ G = \lim_{\alpha_k \underset{\Omega}{\to} \alpha} J\mathcal{L}(\alpha_k) \right\},$$

where conv is the convex hull and $\alpha_k \underset{\Omega}{\to} \alpha$ is a shorthand notation for $\alpha_k \to \alpha$ and $\alpha_k \in \Omega$. It is known (see [20, Proposition 6.2.2]) that $\partial^C \mathcal{L}(\alpha)$ is a nonempty convex compact subset of $\mathbb{R}^r$.

Semi-smooth functions (see [28, Definition 7.4.2]) are precisely (locally) Lipschitz continuous functions for which the Clarke Jacobians define a legitimate Newton approximation scheme in the sense of [28, Definition 7.2.2]. Here, we will even consider an inexact semismooth Newton method, which is detailed in Algorithm 4.

Algorithm 4 Semi-smooth Newton to solve $\mathcal{L}(\alpha) = 0$.

Require: A point $\alpha_0 \in \mathbb{R}^n$
 1: **for all** $k = 0, 1, 2, \ldots$ **do**
 2:    **if** $\mathcal{L}(\alpha_k) = 0$ **then** stop
 3:    **else**
 4:       Select $G_k \in \partial^C \mathcal{L}(\alpha_k)$, compute α_{k+1} such that

$$\mathcal{L}(\alpha_k) + G_k(\alpha_{k+1} - \alpha_k) = e_k,$$

     where $e_k \in \mathbb{R}^r$ is an error term satisfying $\|e_k\| \le \eta_k \|G_k\|$ and $\eta_k \ge 0$
 5:    **end if**
 6: **end for**

It remains now to identify a broad class of convex functions h to which Algorithm 4 applies. A rich family will be provided by semialgebraic functions, i.e., functions whose graph is defined by some Boolean combination of real polynomial equations and inequalities [26]. An even more general family is that of definable functions on an o-minimal structure over $\mathbb{R}$, which corresponds in some sense to an axiomatization of some of the prominent geometrical properties of semialgebraic geometry [72, 25]. A slightly more general notion is that of a tame function, which is a function whose graph has a definable intersection with every bounded box [9, Definition 2]. Given the variety of optimization problems that can be formulated within the framework of o-minimal structures, our convergence result for Algorithm 4 will be stated for tame functions.

PROPOSITION 3.7 (convergence of Algorithm 4). *Consider the situation of Theorem 3.4, where h is in addition a tame function. Then $\mathcal{L}$ is semismooth and all elements of $\partial^C \mathcal{L}(\alpha^\star)$ are nonsingular. In turn there exists $\bar{\eta}$ such that if $\eta_k \le \bar{\eta}$ for every k, there exists a neighborhood of $\alpha^\star$ such that for all α_0 in that neighborhood, the sequence generated by Algorithm 4 is well defined and converges to $\alpha^\star$ linearly. If $\eta_k \to 0$, the convergence is superlinear.*

In particular, if h is semialgebraic and $e_k = 0$, then there exists a rational number $q > 0$ such that

$$\|\alpha_k - \alpha^\star\| = O\left(\exp(-(1+q)^k)\right).$$

The proof is in section B.5.

Proposition 3.7 provides a remarkably fast local convergence guarantee of Algorithm 4 to find the unique zero of $\mathcal{L}$ in (3.5) provided one starts sufficiently close to that zero. If this requirement is not met, the convergence of the algorithm is not ensured anymore. However we can say that $\|\alpha^\star\| \le \beta$, where the radius β can be easily estimated from (B.3). For instance, for the metric $V = P + Q$, by strong convexity

of modulus $c = 1$ (see Theorem 3.4), we have

$$\|\alpha^\star\|^2 /2 \leqslant {}^1\!\big(h^* \circ P^{1/2}\big)(P^{1/2}x) - \inf {}^1\!\big(h^* \circ P^{1/2}\big) + \frac{1}{2} \|x\|_{Q^+}^2 \,.$$

If $0 \in \operatorname{dom}(h)$, we have the bound, valid for any $z \in \mathbb{R}^N$,

$$-h(0) = \inf(h^*) \leq h^* \circ P^{1/2}(p) \leq \tfrac{1}{2} \|z - p\|^2 + h^* \circ P^{1/2}(p) = {}^1\!\big(h^* \circ P^{1/2}\big)(z),$$

where we have written $p = \operatorname{prox}_{h^* \circ P^{1/2}}(z)$. Thus, setting

$$\beta = {}^1\!\big(h^* \circ P^{1/2}\big)(P^{1/2}x) + \frac{1}{2} \|x\|_{Q^+}^2 + h(0),$$

one can initialize Algorithm 4 with α_0 in the ball of radius β. An alternative way is to run, e.g., an accelerated gradient descent (Nesterov or FISTA), initialized with such an α_0, a few iterations on the strongly smooth problem (B.3) in $\mathbb{R}^r$ (recall $r \ll N$), and use the final iterate as an initialization of Algorithm 4. Note that accelerated (FISTA-type) gradient descent is linearly convergent with the optimal rate $1 - \sqrt{\operatorname{cond}^{-1}}$, where $\operatorname{cond} = (1 + \|P^{-1/2}U\|^2)/c$ is the condition number of problem (B.3) (see Theorem 3.4).

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. We start with the general case where h is any function in $\Gamma_0(\mathcal{H})$.

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

$$(3.6) \qquad \operatorname{prox}_h^V(x) = D^{-1/2} \circ \operatorname{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x \mp \alpha^\star D^{-1}u),$$

where $\alpha^\star$ *is the unique root of*

$$(3.7) \qquad \mathcal{L}(\alpha) = \Big\langle u, x - D^{-1/2} \circ \operatorname{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x \mp \alpha D^{-1}u) \Big\rangle + \alpha,$$

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.
- There is a large class of functions for which $\operatorname{prox}_{h \circ D^{-1/2}}$ can be computed either exactly or efficiently. The case of a separable function h will be considered in section 3.3.3, but the computation is efficient even for many nonseparable functions such as the indicator of the simplex and the max function (see Table 3.1), and many others.
- It is of course straightforward to compute $\operatorname{prox}_{h^*}^V$ from $\operatorname{prox}_h^{V^{-1}}$ either using Theorem 3.8, or using this theorem together with Lemma 3.3 and the Sherman–Morrison inversion lemma. Indeed, when $V = D \pm uu^\top$ we have $V^{-1} = D^{-1} \mp vv^\top$, 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 [7] accounts for the proximity operator w.r.t. a metric $V = D + uu^\top$ (diagonal + rank-1), which is extended here to the $V = D \pm uu^\top$ case. Karimi and Vavasis [37] 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 [7]. The results in Theorems 3.4 and 3.8 are far-reaching generalizations that formalize the algorithmic procedure in [37].

3.3.2. Bisection search. We here discuss solving (3.7) via the bisection method in Algorithm 5, since this will allow us to produce a global complexity bound. The key tool is a bound on the values of α given by the following proposition, which is valid even if P is not diagonal.

PROPOSITION 3.10. *For $r = 1$, the root $\alpha^\star$ of (3.7) lies in the set $[-\beta, \beta]$, where*

$$(3.8) \qquad \beta = \|u\| \cdot \left(2\|x\| + \left\| \mathrm{prox}_h^V(0) \right\| \right)$$

and $\mathrm{prox}_h^V(0)$ is a constant (e.g., it is zero if $0 \in \mathrm{argmin}(h)$, as it is for all positively homogeneous functions).

The proof is in section B.4.

PROPOSITION 3.11 (convergence of Algorithm 5). *For any $\epsilon > 0$, Algorithm 5 will produce a point α such that $|\alpha - \alpha^\star| \leq \epsilon$ in $\log_2(\epsilon/(2c\beta))$ steps, where β is as in (3.8), and c is the strong monotonicity modulus given in Theorem 3.4.*

The proof of the above proposition is immediate, since $\mathcal{L}$ is a strongly monotone operator and one dimensional; hence $\mathcal{L}$ is a monotonically increasing function, and thus the bisection method works. Strong monotonicity implies that for all $\alpha \in \mathbb{R}$, $|\mathcal{L}(\alpha)| \geq c|\alpha - \alpha^\star|$.

The bisection procedure is outlined in Algorithm 5; note that later we will provide Algorithm 6, which is a specialization of bisection to a special class of functions h for which we can find the root with zero error (assuming exact arithmetic). Note that a variant of Proposition 3.10 holds when $r > 1$ (see the end of section 3.2.2), but there is no analogue to the bisection method in dimension $r > 1$ since there is no total order.

Algorithm 5 Bisection method to solve $\mathcal{L}(\alpha) = 0$ when $r = 1$.

Require: Tolerance $\epsilon > 0$
1: Compute the bound β from (3.8), and set $k = 0$
2: Set $\alpha_- = -\beta$ and $\alpha_+ = \beta$
3: **for all** $k = 0, 1, 2, \ldots$ **do**
4:     Set $\alpha_k = \frac{1}{2}(\alpha_- + \alpha_+)$
5:     **if** $\mathcal{L}(\alpha_k) > 0$ **then**
6:         $\alpha_+ \leftarrow \alpha_k$
7:     **else**
8:         $\alpha_- \leftarrow \alpha_k$
9:     **end if**
10:    **if** $k > 1$ and $|\alpha_k - \alpha_{k-1}| < \epsilon$ **then**
11:       **return** α_k
12:    **end if**
13: **end for**

3.3.3. 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 3.12 (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*

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

where $\alpha^\star$ is the unique root of

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

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 3.13. *Consider the situation of Corollary* 3.12. *Assume that for $1 \leqslant i \leqslant N$, $\operatorname{prox}_{h_i/d_i}$ is piecewise affine on $\mathbb{R}$ with $k_i \geq 1$ segments, i.e.,*

$$(3.11) \qquad \operatorname{prox}_{h_i/d_i}(x_i) = \begin{cases} a_0^i x_i + b_0^i & \text{if } x_i \leqslant t_1^i, \\ a_j^i x_i + b_j^i & \text{if } t_j^i \leqslant x_i \leqslant t_{j+1}^i, \ j \in \{1, \ldots, k_i\}, \\ a_{k_i+1}^i x_i + b_{k_i+1}^i & \text{if } t_{k_i+1}^i \leqslant x_i, \end{cases}$$

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 $\operatorname{prox}_h^V(x)$ can be obtained exactly using Algorithm 6 *with binary search for step* 3 *in $O(K \log(K))$ steps, where $K = \sum_{i=1}^N k_i$.*

The proof is in section B.6.

Using Proposition 3.13, we derive Algorithm 6.

Algorithm 6 Exact root-finding algorithm for piecewise affine separable proximity operators.

Require: Piecewise affine proximity operator $\operatorname{prox}_{h_i/d_i}(x_i)$, $i = 1, \ldots, N$, as defined in Proposition 3.13.
1: Sort $\widetilde{\boldsymbol{\theta}} := \bigcup_{i=1}^N \{\pm \frac{d_i}{u_i}(x_i - t_j^i) : j = 1, \ldots, k_i\} \subset \mathbb{R}$ into a list $\boldsymbol{\theta} \in \mathbb{R}^{k'}$ with $k' \leqslant K$.
2: Set $\overline{\boldsymbol{\theta}} := [-\infty, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k'}, +\infty]$.
3: Via the bisection method, detect the interval $[\theta_-, \theta_+)$ with adjacent $\theta_-, \theta_+ \in \overline{\boldsymbol{\theta}}$ that contains the root of $\mathcal{L}(\alpha)$.
4: Compute the root $\alpha^\star = -b/a$ where a and b are determined as follows:
5: For all $i = 1, \ldots, N$, define $j_i \in \{0, \ldots, k_i + 1\}$ such that $t_{j_i}^i \leqslant \theta_- < \theta_+ \leqslant t_{j_i+1}^i$, 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 3.14.
- The sign "$\pm$" in Algorithm 6 refers to the two cases of $V = D \pm uu^\top$ from Corollary 3.12.
- Since (3.10) is piecewise affine, a, b in step 5 for the interval $[\theta_-, \theta_+)$ can be determined by

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

where $\theta_- \leqslant \theta'_- < \theta'_+ \leqslant \theta_+$ and $-\infty < \theta'_-$ and $\theta'_+ < +\infty$. (The use of "θ'" avoids "$\mathcal{L}(-\infty)$.")

Remark 3.15.
- The bulk of complexity in Proposition 3.13 lies in locating the appropriate breakpoints. This can be achieved straightforwardly by sorting followed by a bisection search, as advocated, whose worst-case computational complexity 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 seminorms, and many others; see examples hereafter.
- It goes without saying that Corollary 3.12 can be extended to the "block" separable case (i.e., separable in subsets of coordinates).
- It is important to stress the fact that the reasoning underlying Proposition 3.13 and Algorithm 6 extends to a much more general class of proximity operators prox_{h_i}, and hence functions $f_i \in \Gamma_0(\mathbb{R})$. Indeed, assume that h_i is definable (see section 3.2.2 for details on definable functions). Thus arguing as in the proof of Proposition 3.7, we have that prox_{h_i} is also definable. It then follows from the monotonicity lemma [72, Theorem 4.1] that for any $k \in \mathbb{N}$, one can always find a finite partition $(t^i_j)_{1 \leqslant j \leqslant k_i}$ into k_i disjoint intervals such that prox_{h_i} restricted to each nontrivial interval is C^k and strictly increasing or constant. With such a partition, the right-hand side of (3.11) may be nonlinear in x_i but C^k and increasing on the corresponding open interval. Consequently, the first three steps of Algorithm 6, which consist in locating the appropriate interval $[\theta_-, \theta_+)$ that contains the unique root $\alpha^\star$, remain unchanged. If $\alpha^\star \neq \theta_\pm$, only step 5, which computes $\alpha^\star$, has to be changed to any root-finding method of a one-dimensional nonlinear C^k smooth function on (θ_-, θ_+). For instance, we have shown that $\alpha^\star$ is a nondegenerate root ($\mathcal{L}$ is strictly increasing). Therefore, if $k = 2$, then $\mathcal{L} \in C^2((\theta_-, \theta_+))$, and a natural root-finding scheme would be the Newton method, which provides local quadratic convergence to $\alpha^\star$. More generally, if $k \geq 2$, local higher-order convergence rates can be obtained with the Householder's class of methods.
- In view of the previous two remarks, the case of the $\ell_1 - \ell_2$ norm, which is popularly used to promote group sparsity, can be handled by our framework. This example will be considered in more detail in section 3.3.4.

3.3.4. 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

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

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

scalar variable (i.e., finding the unique root of (3.7)).

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 $\mathcal{L}$ in this case is

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

Positive orthant. We now put Proposition 3.13 on a more concrete footing by explicitly covering the case when h represents nonnegativity constraints. Consider $V = D + uu^\top$ and $h = \iota_{\{x:\, x \geqslant 0\}}$. We will calculate

$$(3.12) \qquad \text{prox}_h^{V^{-1}}(x) = \underset{y \geqslant 0}{\text{argmin}}\, \frac{1}{2}\|y - x\|_{V^{-1}}^2.$$

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 that we search for a value of α such that $\alpha = u^\top \lambda^{(\alpha)}$, or in other words, a root of $\mathcal{L}(\alpha) = \alpha - u^\top \lambda^{(\alpha)}$.

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

$$\mathcal{L}(\alpha) = \alpha + \sum_{i=1}^N \left((u_i x_i + \alpha u_i^2)/d_i\right)\chi_i(\alpha),$$

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

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

$$(3.13) \qquad \|x\|_{1,2} = \sum_{b \in \mathscr{B}} \|x_b\|,$$

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

Without of 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 $\mathrm{prox}_{h \circ D^{-1/2}}$, i.e., block soft-thresholding. The latter gives

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

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

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

where $\mathscr{I}(\alpha) = \{b \in \mathscr{B} : \|x_b - \alpha u_b/d_b\| \geq \lambda/d_b\}$. This is a piecewise smooth function, with breakpoints at the values of α where the active support $\mathscr{I}(\alpha)$ changes. To compute the root of α, it is sufficient to locate the two breakpoints where $\mathcal{L}$ changes sign, and then run a fast root-finding algorithm (e.g., Newton's method) on this interval where α is actually C^∞. Denote by $N_\mathscr{B} = \lfloor N/|b| \rfloor$ the number of blocks. There are at most $2N_\mathscr{B}$ breakpoints, and these correspond to the two real roots of $N_\mathscr{B}$ 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 $O(N_\mathscr{B} \log N_\mathscr{B})$. To locate the breakpoints, a simple procedure is a bisection search on the sorted values, and each step necessitates the evaluation of $\mathcal{L}$. This search also costs at most $O(N_\mathscr{B} \log N_\mathscr{B})$ operations (observe that all inner products and norms in $\mathcal{L}$ can be computed once for all). In summary, locating the interval of breakpoints containing the root takes $O(N_\mathscr{B} \log N_\mathscr{B})$ operations, though we believe this complexity could be made linear in $N_\mathscr{B}$ 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*:

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

Algorithm 2 follows the SR1 method [15], 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, as noted by Gould, "[SR1] has now taken its place alongside the BFGS method as the preeminent updating formula." [33].

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-1 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 in [76] that use an approximation to h as well, and therefore take an *explicit* subgradient step.

Algorithm 7 Subroutine 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 τ_{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
15:                                     the Sherman–Morrison formula}

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

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

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

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

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

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-1 update that satisfies 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 [49], we skip updates whenever $\langle s_k, y_k \rangle$ cannot be guaranteed to be nonzero 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 nonconstant 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. This assumption is standard for Newton and quasi-Newton methods if one wants to get provable convergence guarantees. Indeed, one has to assume some nonsingularity assumption for the iterates to be well defined. We can make our strong convexity assumption hold only locally around a minimizer, but our guarantees will also become of local nature. The strong convexity assumption can be weakened to restricted strong convexity when $h = \iota_S$, where $S \subset \mathbb{R}^N$ is a linear subspace. In this case, problem (P) is equivalent to

$$\min_{x \in S} f \circ \mathrm{proj}_S(x).$$

Thus, since $P = (\gamma \tau_{\mathrm{BB2}})^{-1}\mathrm{Id}$ for the 0SR1 and 0BFGS metrics, it follows from (3.4) that $\mathrm{prox}_{\kappa_k h}^{B_k}(x) \in S$. Hence, from (2.3), the quasi-Newton forward-backward sequence satisfies $(x_k)_{k \in \mathbb{N}} \subset S$. In turn, the quasi-Newton vectors s_k and y_k belong to S, i.e., $\forall k \in \mathbb{N}$,

$$s_k = x_k - x_{k-1} \in S \quad \text{and} \quad y_k = \mathrm{proj}_S(\nabla f(\mathrm{proj}_S(x_k))) - \mathrm{proj}_S(\nabla f(\mathrm{proj}_S(x_{k-1}))) \in S.$$

Now, assuming that h is strongly convex on S and its gradient is Lipschitz on S, with constants μ_S and L_S, the bounds on the eigenvalues of matrices H_k in Lemmas 4.1 and 5.1 hereafter will remain true with (μ, L) replaced by (μ_S, L_S). The convergence claims of Theorems 4.2 and 5.2 will also hold with rates characterized by the condition number L_S/μ_S rather than L/μ.

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, $\forall k \geq 0$, $a\mathrm{Id} \preceq H_k \preceq b\mathrm{Id}$, $0 < a = \gamma L^{-1}$, and*

$$0 < b = \frac{(1+\gamma)\mu^{-1} - 2\gamma L^{-1}}{1 - \gamma}.$$

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 $0 < \underline{\kappa} \leqslant \kappa_k \leqslant \overline{\kappa} < 2(Lb)^{-1}$. Let*

$$\alpha = 1 - \frac{Lb\overline{\kappa}}{2} \quad \text{and} \quad \eta = \frac{L}{2\gamma\mu\underline{\kappa}}.$$

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 $x^\star$, i.e.,

$$\|x_k - x^\star\| \leqslant \sqrt{\frac{2\left(F(x_0) - F(x^\star)\right)}{\mu}}\rho^{k/2},$$

where

$$\rho = \begin{cases} \rho_1 & \text{for } \alpha \in \,]0, 1/2[, \\ \min(\rho_1, \rho_2) & \text{for } \alpha \in [1/2, 1[, \end{cases}$$
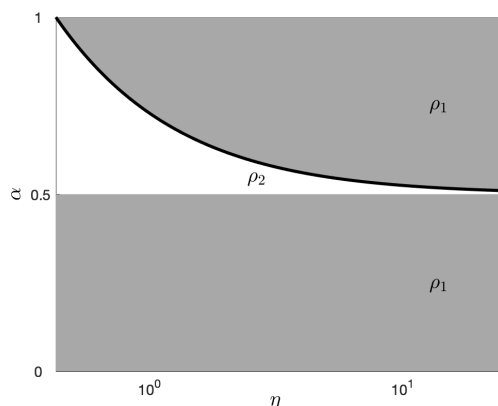
FIG. 4.1. *Convergence rate as a function of the parameters η and α (see Theorem 4.2 for details).*

with

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

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

Actually, this is the standard setting for Newton and quasi-Newton methods if one wants to get provable convergence guarantees. Indeed, one has to assume some nonsingularity assumption for the iterates to be well defined. We can make our strong convexity assumption hold only locally around a minimizer, but our guarantees will also become of local nature.

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 $\kappa_k \equiv \overline{\kappa} = \underline{\kappa} = 1/(Lb)$. 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 $\rho_1 - \rho_2 \to 0$ as $\eta \to \infty$. In turn, the rate of linear convergence is $\rho \approx 1 - 1/(8\eta) \approx 1 - 1/(24c^2)$. Although this rate is apparently worse than, for example, the standard rate obtained for forward-backward, our numerical experiments demonstrate that the performance is significantly better than this worst-case prediction. Unless the metric approximates second-order information, which is not the case for our zero-memory variant, we do not expect to improve the convergence rate. Possibly, a deep analysis might improve the constants appearing in the convergence rate estimate. However, the efficiency of our method comes from an "optimal" compromise between locally adapting the metric and a cheap computability of the update step.

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 [49]. The proximity operator in (2.3) will be of type "diagonal $\pm$ rank-2."

5.1. Metric construction. Define

$$\rho_k = \frac{1}{y_k^\top s_k}, \quad V_k = \mathrm{Id} - \rho_k y_k s_k^\top, \quad \text{with} \quad s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k),$$

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

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

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 $\tau = \tau_{\mathrm{BB2}}$, writing V for V_k and so on, we can expand as

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

(5.1)

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

(5.2) $$H_k = \tau\mathrm{Id} + 2\rho\Big(s - \frac{\tau}{2}y\Big)\Big(s - \frac{\tau}{2}y\Big)^\top - \rho\frac{\tau^2}{2} y y^\top.$$

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

$$\begin{aligned}
B_k & = B_k^0 - \frac{B_k^0 s s^\top B_k^0}{s^\top B_k^0 s} + \frac{y y^\top}{y^\top s} = \frac{1}{\gamma\tau}\Big(\mathrm{Id} - \frac{s s^\top}{s^\top s} + \gamma\tau\frac{y y^\top}{y^\top s}\Big) \\
& \overset{(\tau = \tau_{\mathrm{BB2}})}{=} \frac{1}{\gamma\tau_{\mathrm{BB2}}}\Big(\mathrm{Id} - \frac{s s^\top}{s^\top s} + \gamma\frac{y y^\top}{y^\top y}\Big).
\end{aligned}$$

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 $\mathrm{prox}_h^{B_0}$ is required. The main computational cost comes from the two-dimensional root-finding problem, which can be solved efficiently using semismooth Newton methods.

5.2. Convergence analysis. For the convergence analysis, we again assume that f is also μ-strongly convex. We start with a lemma which provides 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, $\forall k \geq 0$, $a\mathrm{Id} \preceq H_k \preceq b\mathrm{Id}$, $0 < a = \gamma/(1+\gamma)L^{-1}$, and $0 < b = (1+2\gamma)\mu^{-1} - \frac{(2+\gamma)\gamma}{1+\gamma}L^{-1}$.*

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 $0 < \underline{\kappa} \leqslant \kappa_k \leqslant \overline{\kappa} < 2(Lb)^{-1}$. Let $\alpha = 1 - \frac{Lb\overline{\kappa}}{2}$ 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 $x^\star$, i.e.,*

$$\|x_k - x^\star\| \leqslant \sqrt{\frac{2\left(F(x_0) - F(x^\star)\right)}{\mu}}\rho^{k/2},$$

where ρ is as given in Theorem 4.2.

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 for those in Lemma 5.1. Note that the phase diagram in Figure 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 $\kappa_k \equiv \overline{\kappa} = \underline{\kappa} = 1/(Lb)$. 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 0SR1 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 online.[1]

6.1. The LASSO problem. Consider the unconstrained LASSO problem (1.1). Many codes, such as [27] and L-BFGS-B [16], handle only nonnegativity or box constraints. Using the standard change of variables by introducing the positive and negative parts of x, the LASSO can be recast as

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

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

---

[1]See https://github.com/stephenbeckr/zeroSR1/tree/master/paperExperiments.

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 ± rank-1 proximal mapping. Our implementation of FISTA used the Barzilai–Borwein step size* [4] *and line search, and restarted the momentum term every* 1000 *iterations* [3]. *L-BFGS-B and ASA use the reformulation of* (6.1). *For L-BFSG-B, we use the updated version* [45]. *The code for PSSas and OWL (a slight variant of* [1]*) is from* [63].

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 [75]	§6.1, 6.2	MATLAB version from [75]
L-BFGS-B	Limited memory, box-constrained BFGS [16, 77]	§6.1	Fortran with MATLAB wrapper
ASA	Active-set algorithm (conjugate gradient) [36]	§6.1	C with MATLAB wrapper, ver. 2.2
OWL	Orthant-wise learning [1]	§6.1	Active set; MATLAB
PSSas	Projected scaled subgradient + active set [64]	§6.1	MATLAB
CGIST	CG + IST [32]	§6.1	MATLAB
FPC-AS	Fixed-point continuation + active set [74]	§6.1	MATLAB, ver. 1.21
0-mem SR1	Algorithm 7	§6.1, 6.2	**Our approach** (in MATLAB)

constraints $x_+ \geqslant 0$, $x_- \geqslant 0$, and replace $h(x)$ with $\mathbf{1}^\top(x_+)$. As before, the Hessian gains n degenerate eigenvalues.

We compared our proposed algorithm applied to the LASSO problem to several algorithms of the literature, as summarized in Table 6.1. The first example, on the left of Figure 6.1,[2] is a typical example from compressed sensing that takes $A \in \mathbb{R}^{m \times n}$ to have independently and identically distributed $\mathcal{N}(0,1)$ 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 three-dimensional discrete differential operator. This example stems from a numerical analysis problem of solving a discretized PDE as suggested by [30]. 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 (3.13), 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 $b \in \mathscr{B}$ with randomly selected size $|b| \leqslant 12$. For

---

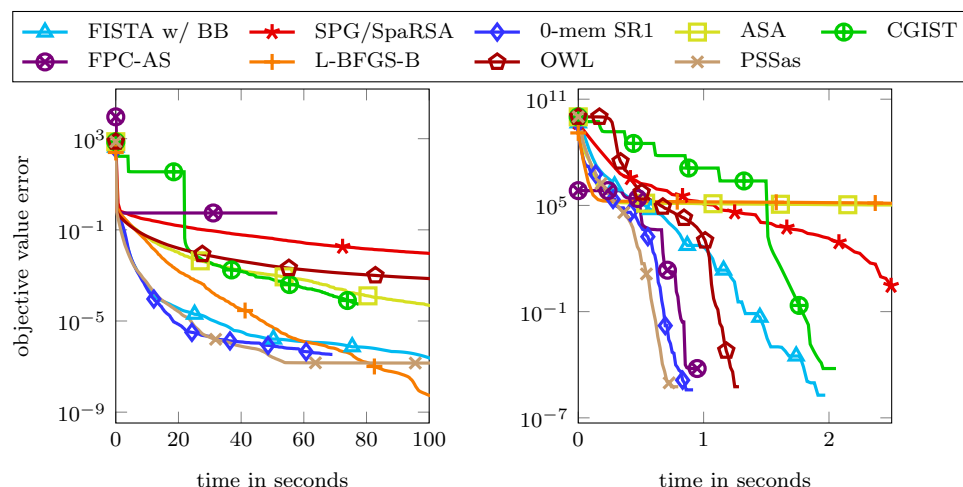[2]See online version for color figures.

FIG. 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.*

the numerical experiment, the entries of A and b are drawn uniformly in $[0, 1]$, and we set $N = 2500$, $M = 1600$, 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 3.3.4. We apply Newton's method in the interval between breakpoints that locates the root.

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 the best performance in the low and medium precision regime, while, for obtaining a high precision, accelerated strategies, such as FISTA, seem to be favorable. Presumably, this comes from the $\ell_1 - \ell_2$ norm, which usually activates a whole block of coordinates, unlike in the LASSO case where eventually only a few coordinates are active and it thus often has an improved condition number when restricted to these active variables. Acceleration strategies seem to compensate for this effect. In the beginning, the SR1 metric reflects the conditioning of the problem better than isotropic metrics.

Figure 6.2 also suggests that the improvement with respect to FISTA could be further increased when a more efficient implementation of the diagonal $\pm$ rank-1 proximal mapping is used, or when the rank-1 update is combined with the acceleration strategy as in [52], which we will explore in future work.

7. Conclusions. In this paper, we proposed a novel framework for variable metric (quasi-Newton) forward-backward splitting algorithms, designed to efficiently solve nonsmooth convex problems structured as the sum of a smooth term and a nonsmooth 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 [7], and we believe it is of independent interest as even the simpler version from [7] has been the basis of other works such as [37, 52]. We
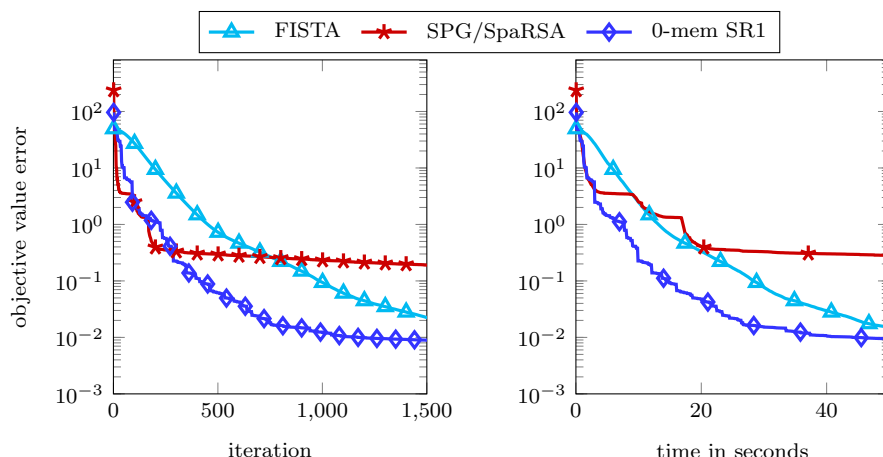
FIG. 6.2. *Convergence plots for the methods described in section* 6.2 *for solving the* $\ell_1 - \ell_2$ *LASSO problem. The vertical axis is the same for both plots. The methods based on the efficient solution of the diagonal* $\pm$ *rank-*1 *proximal mapping proposed in this paper outperform comparable methods based on a diagonally scaled proximal mapping.*

also established convergence of the algorithm, and provided clear evidence that the nondiagonal 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 [59]. However, if we switch to a primal-dual setting, which is desirable because it can handle more complicated objective functionals, updating B_k is not obvious, though one could perhaps use our results for a nondiagonal preconditioning method.

Another improvement would be to derive an efficient calculation for the 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 semismooth 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 can still be solved by our algorithm (see Proposition 3.13). 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 [35], 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 [61, 5].

A.1. Background.

Functions.

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

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

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

DEFINITION A.2 (infimal convolution). *Let h_1 and h_2 be 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*

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

Conjugacy.

DEFINITION A.3 (conjugate). *Suppose that $h : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ has a minorizing affine function. The conjugate or Legendre–Fenchel transform of h on $\mathcal{H}$ is the function h^* defined by*

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

LEMMA A.4 (calculus rules).
 (i) $(h(x) + t)^*(v) = h^*(v) - t$.
 (ii) $(th(x))^*(v) = tf^*(v/t)$, $t > 0$.
 (iii) $(h \circ A)^* = h^* \circ \left(A^{-1}\right)^*$ *if A is a linear invertible operator.*
 (iv) $(h(x - x_0))^*(v) = h^*(v) + \langle v, x_0 \rangle$.
 (v) *Separability:* $\left(\sum_{i=1}^n h_i(x_i)\right)^*(v_1, \dots, v_n) = \sum_{i=1}^n h_i^*(v_i)$, *where* $(x_1, \dots, x_n) \in \mathcal{H}_1 \times \cdots \times \mathcal{H}_n$.
 (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*

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

 (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 semidefinite matrix. Let Q^+ be its Moore–Penrose pseudoinverse. Then,*

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

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 $\partial h : \mathcal{H} \to 2^{\mathcal{H}}$,*

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

An element v of ∂h is called a subgradient.

The subdifferential map ∂h is a maximal monotone operator from $\mathcal{H} \to 2^{\mathcal{H}}$.

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 $0 \in \mathrm{ri}\,(\mathrm{dom}\,g - \mathrm{dom}\,h)$. Then*

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

and the solutions of the primal and dual problems, $x^\star$ and $u^\star$, respectively, satisfy

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

Toland duality.

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

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

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

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

Proof. The first assertion is a consequence of [69, Theorem 2.2]. The extremality relationships follow by combining [69, Theorems 2.7 and 2.8]. □

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

DEFINITION A.12 (Moreau envelope [46]). *The function*

$$^\rho h(x) = \inf_{z \in \mathcal{H}} \frac{1}{2\rho} \left\| x - z \right\|^2 + h(z)$$

for $0 < \rho < +\infty$ is the Moreau envelope *of index ρ of h.*

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

LEMMA A.13.
  (i) *Translation:* $\mathrm{prox}_{h(\cdot - y)}(x) = y + \mathrm{prox}_h(x - y)$.
  (ii) *Scaling:* $\forall \rho \in (-\infty, \infty)$, $\mathrm{prox}_{h(\rho \cdot)}(x) = \mathrm{prox}_{\rho^2 f}(\rho x)/\rho$.
  (iii) *Separability: let $(h_i)_{1 \le i \le n}$ 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 $\mathrm{prox}_h = \left(\mathrm{prox}_{h_i}\right)_{1 \le i \le N}$.*

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*

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

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

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

From Lemma A.15, we conclude that

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

Appendix B. Proofs of section 3.

B.1. Proof of Lemma 3.2.

Proof. Let $p = \mathrm{prox}_h^V(x)$. The statement follows from the following equivalences

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

B.2. Proof of Lemma 3.3.

Proof. We have

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

B.3. Proof of Theorem 3.4.

Proof. Let $p = \mathrm{prox}_h^V(x)$. Then, we have to solve

$$
\min_z \frac{1}{2}\|x - z\|_V^2 + h(z)
$$

$$
\Leftrightarrow \min_z \left(\frac{1}{2}\|z\|_P^2 - \langle x, z\rangle_P + h(z)\right) \pm \frac{1}{2}\langle x-z, Q(x-z)\rangle
$$

$$
\begin{bmatrix} y = P^{1/2}z \\ W = P^{-1/2}QP^{-1/2} \end{bmatrix} \Leftrightarrow \min_y \left(\frac{1}{2}\|y\|^2 - \left\langle P^{1/2}x, y\right\rangle + h \circ P^{-1/2}(y)\right)
$$

$$
(\mathrm{B.1}) \hspace{4cm} \pm \frac{1}{2}\left\langle P^{1/2}x - y, W(P^{1/2}x - y)\right\rangle
$$

$$
\begin{bmatrix} \text{Lemma A.10(A.1)} \\ \text{or Lemma A.11(A.3)} \end{bmatrix} \Leftrightarrow \min_w \pm \left(\frac{1}{2}\|\cdot\|^2 - \left\langle P^{1/2}x, \cdot\right\rangle + h \circ P^{-1/2}\right)^* (\mp w)
$$

$$
\hspace{3cm} + \left(\frac{1}{2}\left\langle P^{1/2}x - \cdot, W(P^{1/2}x - \cdot)\right\rangle\right)^* (w)
$$

$$
\begin{bmatrix} \text{Lemma A.5} \\ \text{and Lemma A.4(iv)} \end{bmatrix} \Leftrightarrow \min_{w \in \mathrm{Im}(W)} \pm \left(\frac{1}{2}\|\cdot\|^2 - \left\langle P^{1/2}x, \cdot\right\rangle + h \circ P^{-1/2}\right)^* (\mp w)
$$

$$
\hspace{3cm} + \frac{1}{2}\|w\|_{W^+}^2 + \left\langle P^{1/2}x, w\right\rangle
$$

$$
[\text{Lemma A.4(vi)-(iii)}] \Leftrightarrow \min_{w \in \mathrm{Im}(W)} \pm \left(\left(\frac{1}{2}\|\cdot\|^2 - \left\langle P^{1/2}x, \cdot\right\rangle\right)^* \overset{+}{\vee} (h^* \circ P^{1/2})\right)(\mp w)
$$

$$
\hspace{3cm} + \frac{1}{2}\|w\|_{W^+}^2 + \left\langle P^{1/2}x, w\right\rangle
$$

$$
\Leftrightarrow \min_{w \in \mathrm{Im}(W)} \pm \left(\left(\frac{1}{2}\left\|P^{1/2}x + \cdot\right\|^2\right) \overset{+}{\vee} (h^* \circ P^{1/2})\right)(\mp w)
$$

$$
\hspace{3cm} + \frac{1}{2}\|w\|_{W^+}^2 + \left\langle P^{1/2}x, w\right\rangle
$$

$$\text{[Definition A.12]} \Leftrightarrow \min_{w \in \text{Im}(W)} \pm^{1}\Big(h^* \circ P^{1/2}\Big)(P^{1/2}x \mp w)$$

$$\text{(B.2)} \qquad\qquad + \frac{1}{2}\|w\|^2_{W^+} + \Big\langle P^{1/2}x, w \Big\rangle.$$

By virtue of Lemma A.14, $^{1}\big(h^* \circ P^{1/2}\big)$ is continuously differentiable with 1-Lipschitz gradient. Together with Lemma A.8, Lemma A.10(A.2) or Lemma A.11(A.4),[3] and Lemma A.15, this yields

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

where $w^\star$ 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

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

Since the columns of U are linearly independent, $U^\top Q^+ U$ is nothing but the identity operator on $\mathbb{R}^r$. The gradient of the objective in (B.3) is given by the mapping $\mathcal{L}$. Lipschitz continuity of $\mathcal{L}$ follows from nonexpansiveness of the proximal mapping, and the Lipschitz constant is straightforward from the triangle and Cauchy–Schwartz inequality. The root $\alpha^\star$ of $\mathcal{L}$ is unique if $\mathcal{L}$ is strongly monotone. In the $V = P+Q$ case, strong monotonicity is immediate since all terms in (B.3) are convex, and $\|\alpha\|^2_{U^\top Q^+ U}$ is strongly convex of modulus 1.

In the $V = P - Q$ case, 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, [24, Lemma 2.10]) to the first term, which reduces the analysis of strong convexity to that of $\big\langle \alpha, (U^\top(Q^+ - P^{-1})U)\alpha \big\rangle$, and hence, the positive definiteness of $U^\top(Q^+ - P^{-1})U$. 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

$$1 = \|AA^{-1}\| \leqslant \|A\|\|A^{-1}\|$$

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 $Q^+ - P^{-1} \in \mathbb{S}_{++}(\text{Im}(Q))$ and, thus, (B.3) is strongly convex. Its modulus of strong convexity is

$$\|U^\top(Q^+ - P^{-1})U\| = 1 - \|U^\top P^{-1}U\| = 1 - \|P^{-1/2}U\|^2. \qquad \square$$

B.4. Proof of Proposition 3.10.

Proof. We use the notation of Theorem 3.4 and its proof in Appendix B.3. Let $p = \text{prox}_h^V(x)$. By nonexpansivity of the proximal mapping, $\|p\| = \big\|\text{prox}_h^V(x)\big\| \leqslant$

---

[3]The coercivity assumption holds (in fact the primal has exactly one solution) and the dual problem has indeed a nonempty set of minimizers.

$\|x\| + \left\|\mathrm{prox}_h^V(0)\right\|$. Since p minimizes $\frac{1}{2}\|x-z\|_V^2 + h(z)$, and using the same change of variable $y = P^{1/2}z$ as in the proof, the optimal point satisfies $y^\star = P^{1/2}p$.

Letting $g(y) = \pm\frac{1}{2}\left\langle P^{1/2}x - y, W(P^{1/2}x - y)\right\rangle$ with $W = P^{-1/2}QP^{-1/2}$, either Lemma A.10 or Lemma A.11 gives the optimal dual solution

$$
\begin{aligned}
\mp w^\star &= \nabla g(y^\star)\\
&= W(y^\star - P^{1/2}x)\\
&= P^{-1/2}QP^{-1/2}y^\star - P^{-1/2}Qx\\
&= P^{-1/2}Q(p-x).
\end{aligned}
$$

(B.4)

Finally, $w^\star = P^{-1/2}U\alpha^\star$, and observe $U = u$ since $r = 1$, and so also $Q = uu^\top$. Then

$$
\begin{aligned}
|\alpha^\star| &= \left\|P^{1/2}w^\star\right\|/\|u\|\\
&= \|Q(p-x)\|/\|u\| \quad \text{(via (B.4))}\\
&\leq \|u\|\left(2\|x\| + \left\|\mathrm{prox}_h^V(0)\right\|\right). \qquad \square
\end{aligned}
$$

B.5. Proof of Proposition 3.7.

Proof. The key of the proof is the remarkable stability properties of definable functions. In particular, under the sum, composition by a linear operator, derivation, and canonical projection (see [72, 25]). Since h is a tame function, so is $h \circ P^{-1/2}$, as well as its Moreau envelope (by the projection stability), and the gradient of the latter. Combining this with Lemma A.14, it follows that $\mathrm{prox}_{h\circ P^{-1/2}}$ is a tame mapping. We then deduce from stability to the sum and composition by a linear operator that $\mathcal{L}$ is a tame mapping. Thus, $\mathcal{L}$ is a tame Lipschitz continuous mapping (Theorem 3.4), and it follows from [9, Theorem 1] that $\mathcal{L}$ is semismooth.

Let us now show that $\partial^C\mathcal{L}(\alpha^\star)$ is nonsingular. By definition of the Clarke Jacobian for a Lipschitz function and the Carathéodory theorem, for any $G \in \partial^C\mathcal{L}(\alpha^\star)$, we have a finite sequence $\rho_1,\ldots,\rho_{r^2+1} \geq 0$ living on the simplex, i.e., $\sum_{i=1}^{r^2+1}\rho_i = 1$, and $r^2 + 1$ sequences $(\alpha_{i,k})_{k\in\mathbb{N}}$ with $\alpha_{i,k} \underset{\Omega}{\to} \alpha^\star$ as $k \to +\infty$ such that for any $d \in \mathbb{R}^r$,

$$
\begin{aligned}
\langle Gd, d\rangle &= \sum_{i=1}^{r^2+1}\rho_i \lim_{k\to+\infty}\langle J\mathcal{L}(\alpha_{i,k})d, d\rangle\\
&= \sum_{i=1}^{r^2+1}\rho_i \lim_{k\to+\infty}\lim_{\tau\to 0}\frac{\langle\mathcal{L}(\alpha_{i,k}+\tau d) - \mathcal{L}(\alpha_{i,k}), d\rangle}{\tau}.
\end{aligned}
$$

By strong monotonicity of $\mathcal{L}$ of modulus $c > 0$ (Theorem 3.4), we have for all $d \in \mathbb{R}^r$,

$$
\frac{\langle\mathcal{L}(\alpha_{i,k}+\tau d) - \mathcal{L}(\alpha_{i,k}), d\rangle}{\tau} \geq c\|d\|^2.
$$

Passing to the limit and summing, we conclude that

$$
\langle Gd, d\rangle \geq c\|d\|^2 \quad \forall d \in \mathbb{R}^r.
$$

Since G is any element of $\partial^C\mathcal{L}(\alpha^\star)$, we get that $\partial^C\mathcal{L}(\alpha^\star)$ is nonsingular. We are then in position to apply [28, Theorem 7.5.5] to obtain the first part of the convergence claim.

For the case where h is semialgebraic, we argue as above, using stability of semialgebraic sets to the same operations (in particular projection stability by the Tarski–Seidenberg principle [26]), to deduce that $\mathrm{prox}_{h \circ P^{-1/2}}$ is also a semialgebraic mapping. The last claim then follows from [9, Theorem 2]. $\qquad\square$

B.6. Proof of Proposition 3.13.

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 $\mathcal{L}(\alpha)$ in (3.10) is also piecewise affine with slopes and intercepts changing at the k' (unique) transition points $\widetilde{\boldsymbol{\theta}}$. Therefore, the root of $\mathcal{L}$ can be found by sorting $\widetilde{\boldsymbol{\theta}}$ (step 1 of Algorithm 6) 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 $\mathcal{L}$, which consists of N terms. Step 5 adds at most a complexity of $O(N)$. $\qquad\square$

Appendix C. Proofs of section 4.

C.1. Proof of Lemma 4.1.

Proof. From [48, pp. 57 and 64], we have for any x and y in $\mathrm{dom}(F)$,

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

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

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

We will use the "2nd" Barzilai–Borwein step size τ_{BB2} as opposed to the more common τ_{BB1}:

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

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

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

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

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

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

Thus

$$0 \prec \gamma L^{-1} \mathrm{Id} \preceq H_0 \preceq H_k \preceq \gamma \mu^{-1} \mathrm{Id} + (1-\gamma)^{-1} \left((1+\gamma^2)\mu^{-1} - 2\gamma L^{-1} \right) \mathrm{Id}$$
$$\preceq (1-\gamma)^{-1} \left((1+\gamma)\mu^{-1} - 2\gamma L^{-1} \right) \mathrm{Id}. \qquad \square$$

C.2. Proof of Theorem 4.2.

Proof. We first recall the classical inequality for smooth functions with L-Lipschitz continuous gradient,

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

The $\alpha \in]0, 1/2[$ case. 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

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

Applied at x_k, this yields

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

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

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

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

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

By assumption, the right-hand side is nonpositive, meaning that the objective function decreases with k. Define

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

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 $y = x^\star$, we obtain

$$E_k = f(x_k) - f(x^\star) + \langle \nabla f(x_k), x^\star - x_k \rangle$$
$$+ h(x_k) - h(x_{k+1}) + \langle \nabla f(x_k), x_k - x_{k+1} \rangle$$
$$+ h(x_{k+1}) - h(x^\star) + \langle \nabla f(x_k), x_{k+1} - x^\star \rangle$$
$$\leqslant D_k + \kappa_k^{-1} \langle B_k(x_k - x_{k+1}), x_{k+1} - x^\star \rangle$$
$$\leqslant D_k + \kappa_k^{-1} \|x_{k+1} - x^\star\|_{B_k} \|x_{k+1} - x_k\|_{B_k}$$
$$\leqslant D_k + \sqrt{\frac{1}{\underline{\kappa}a}} \|x_{k+1} - x^\star\| \sqrt{D_k}$$
$$\leqslant \alpha^{-1} \left(\Delta_k + \sqrt{\frac{\alpha}{\underline{\kappa}a}} \|x_{k+1} - x^\star\| \sqrt{\Delta_k} \right).$$

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

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

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

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

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

$$
\left\|x_k - x^\star\right\| \leqslant \sqrt{\frac{2\left(F(x_0) - (Fx^\star)\right)}{\mu}}\rho^{k/2}.
$$

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.

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

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

Moreover, convexity of f yields

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

It then follows that

$$
\begin{aligned}
F(x_{k+1}) &\leqslant Q_k^{B_k}(x_{k+1}) + h(x_{k+1}) \\
&\leqslant \min_x F(x) + \frac{1}{2\kappa_k}\left\|x - x_k\right\|_{B_k}^2
\end{aligned}
$$

$$
\leqslant \min_{t \in [0,1]} F(tx^\star + (1-t)x_k) + \frac{t^2}{2\kappa_k} \|x_k - x^\star\|_{B_k}^2
$$

$$
\leqslant \min_{t \in [0,1]} tF(x^\star) + (1-t)F(x_k) + \frac{t^2 L}{2\underline{\kappa}\gamma} \|x_k - x^\star\|^2
$$

$$
\leqslant \min_{t \in [0,1]} F(x_k) - t\left(F(x_k) - F(x^\star)\right) + \frac{t^2 L}{\underline{\kappa}\gamma\mu}\left(F(x_k) - F(x^\star)\right)
$$

$$
= \min_{t \in [0,1]} F(x_k) - t\left(1 - 2t\eta\right)\left(F(x_k) - F(x^\star)\right).
$$

Thus, we arrive at

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

The function $\left(1 - t\left(1 - 2t\eta\right)\right)$ attains its minimum uniquely at $t = 1$ if $\eta \leqslant 1/4$, and $1/(4\eta)$ otherwise. Plugging these values in gives the expression for ρ_2. □

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), with $\tau = \tau_{\mathrm{BB2}}$, satisfies

$$
\rho \|u_\gamma\|^2 = \frac{\|s\|^2 - 2\frac{\gamma\tau}{1+\gamma}\langle y, s\rangle + \left(\frac{\gamma\tau}{1+\gamma}\right)^2 \|y\|^2}{\langle y, s\rangle} = \frac{\|s\|^2}{\langle y, s\rangle} - 2\frac{\gamma}{1+\gamma}\tau_{\mathrm{BB2}} + \frac{\gamma^2}{(1+\gamma)^2}\tau_{\mathrm{BB2}}
$$

$$
= \tau_{\mathrm{BB1}} - \frac{\gamma}{1+\gamma}\left(2 - \frac{\gamma}{1+\gamma}\right)\tau_{\mathrm{BB2}} \leqslant \mu^{-1} - \frac{(2+\gamma)\gamma}{(1+\gamma)^2}L^{-1},
$$

where we used $\rho^{-1} = \langle y, s\rangle$, $\rho \|y\|^2 = \tau_{\mathrm{BB2}}^{-1}$, and the estimations in the proof of Lemma 4.1 for τ_{BB2} and τ_{BB1}. Using this estimation, $\rho\tau_{\mathrm{BB2}}^2 \|y\|^2 = \tau_{\mathrm{BB2}} \geq L^{-1}$, and positive semidefiniteness of $yy^\top$ and $u_\gamma u_\gamma^\top$, we conclude that

$$
0 \prec \frac{\gamma}{1+\gamma}L^{-1}\mathrm{Id} = \gamma\left(\tau_{\mathrm{BB2}} - \frac{\gamma^2}{1+\gamma}\tau_{\mathrm{BB2}}\right)\mathrm{Id}
$$

$$
\preceq H
$$

$$
\preceq \tau_{\mathrm{BB2}}\gamma\mathrm{Id} + (1+\gamma)\mu^{-1}\mathrm{Id} - \frac{(2+\gamma)\gamma}{1+\gamma}L^{-1}\mathrm{Id}
$$

$$
\preceq (1+2\gamma)\mu^{-1}\mathrm{Id} - \frac{(2+\gamma)\gamma}{1+\gamma}L^{-1}\mathrm{Id}. \qquad □
$$

REFERENCES

[1] G. ANDREW AND J. GAO, *Scalable training of L^1-regularized log-linear models*, in Proceedings of the 24th International Conference on Machine Learning, ACM Press, New York, NY, 2007, pp. 33–40.

[2] J. ATTOUCH AND H. PEYPOUQUET, *The rate of convergence of Nesterov's accelerated forward-backward method is actually faster than $1/k^2$*, SIAM J. Optim., 26 (2016), pp. 1824–1834.

[3] B. B. O'DONOGHUE AND E. CANDÈS, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., 15 (2015), pp. 715–732.

[4] J. BARZILAI AND J. BORWEIN, *Two point step size gradient method*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[5] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math./Ouvrages Math. SMC, Springer, Cham, 2011.

[6] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[7] S. BECKER AND J. FADILI, *A quasi-Newton proximal splitting method*, in in Adv. Neural Inf. Process. Syst. 25, Curran Associates, Red Hook, NY, 2012, pp. 2618–2626.

[8] E. G. BIRGIN, J. M. MARTÍNEZ, AND M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim., 10 (2000), pp. 1196–1211.

[9] J. BOLTE, A. DANIILIDIS, AND A. LEWIS, *Tame functions are semismooth*, Math. Program., 117 (2009), pp. 5–19.

[10] S. BONETTINI, I. LORIS, F. PORTA, AND M. PRATO, *Variable metric inexact line-search based methods for nonsmooth optimization*, SIAM J. Optim., 26 (2016), pp. 891–921.

[11] 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*, preprint, https://arxiv.org/abs/1605.03791, 2016.

[12] S. BONETTINI AND M. PRATO, *New convergence results for the scaled gradient projection method*, Inverse Problems, 31 (2015), No. 9.

[13] S. BONETTINI, R. ZANELLA, AND L. ZANNI, *A scaled gradient projection method for constrained image deblurring*, Inverse Problems, 25 (2009), No. 1.

[14] K. BREDIES AND H. SUN, *Preconditioned Douglas–Rachford splitting methods for convex-concave saddle-point problems*, SIAM J. Numer. Anal., 53 (2015), pp. 421–444.

[15] C. BROYDEN, *Quasi-Newton methods and their application to function minimization*, Math. Comp., 21 (1967), pp. 577–593.

[16] R. H. BYRD, P. LU, J. NOCEDAL, AND C. ZHU, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.

[17] G. H.-G. CHEN AND R. T. ROCKAFELLAR, *Convergence rates in forward–backward splitting*, SIAM J. Optim., 7 (1997), pp. 421–444.

[18] X. CHEN, Z. NASHED, AND L. QI, *Smoothing methods and semismooth methods for nondifferentiable operator equations*, SIAM J. Numer. Anal., 38 (2000), pp. 1200–1216.

[19] 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*, J. Optim. Theory Appl., 162 (2014), pp. 107–132.

[20] F. CLARKE, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, PA, 1990.

[21] P. L. COMBETTES AND B. C. VŨ, *Variable metric quasi-Fejér monotonicity*, Nonlinear Anal., 78 (2013), pp. 17–31.

[22] P. L. COMBETTES AND B. C. VŨ, *Variable metric forward-backward splitting with applications to monotone inclusions in duality*, Optimization, 63 (2014), pp. 1289–1318.

[23] P. L. COMBETTES AND J. C. PESQUET, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, eds., Springer, New York, 2011, pp. 185–212.

[24] P. L. COMBETTES AND V. R. WAJS, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200.

[25] M. COSTE, *An Introduction to O-minimal Geometry*, Technical report, Institut de Recherche Mathematiques de Rennes, Rennes, 1999.

[26] M. COSTE, *An Introduction to Semialgebraic Geometry*, Technical report, Institut de Recherche Mathematiques de Rennes, Rennes, 2002.

[27] I. DHILLON, D. KIM, AND S. SRA, *Tackling box-constrained optimization via a new projected quasi-Newton approach*, SIAM J. Sci. Comput., 32 (2010), pp. 3548–3563.

[28] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vol. I, Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2003.

[29] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vol. II, Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2003.

[30] R. FLETCHER, *On the Barzilai–Borwein method*, in Optimization and Control with Applications, L. Qi, K. Teo, X. Yang, P. Pardalos, and D. W. Hearn, eds., Appl. Optim. 96, Springer, New York, 2005, pp. 235–256.

[31] M. P. FRIEDLANDER AND G. GOH, *Efficient evaluation of scaled proximal operators*, Electron. Trans. Numer. Anal., 46 (2017), pp. 1–22.

[32] T. GOLDSTEIN AND S. SETZER, *High-order Methods for Basis Pursuit*, Technical report, University of California, Los Angeles, CA, 2011.

[33] N. GOULD, *Seminal papers in nonlinear optimization*, in An Introduction to Algorithms

for Continuous Optimization, Oxford University Computing Laboratory and Rutherford Appleton Laboratory, 2006; available at http://www.cs.ox.ac.uk/nick.gould/courses/co/paper.pdf.

[34] J. GUO AND A. LEWIS, *BFGS Convergence to Nonsmooth Minimizers of Convex Functions*, preprint, https://arxiv.org/abs/1703.06690, 2017.

[35] W. HACKBUSCH, *A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices*, Computing, 62 (1999), pp. 89–108.

[36] W. W. HAGER AND H. ZHANG, *A new active set algorithm for box constrained optimization*, SIAM J. Optim., 17 (2006), pp. 526–557.

[37] S. KARIMI AND S. VAVASIS, *IMRO: A proximal quasi-Newton method for solving ℓ_1-regularized least squares problems*, SIAM J. Optim., 27 (2017), pp. 583–615.

[38] M. KOJIMA AND S. SHINDO, *Extension of Newton and quasi-Newton methods to systems of PC^1 equations*, J. Oper. Res. Soc. Japan, 29 (1986), pp. 352–375.

[39] B. KUMMER, *Newton's method for non-differentiable functions*, in Advances in Mathematical Optimization, J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Klatte, B. Kummer, K. Lommatzsch, L. Tammer, M. Vlach, and K. Zimmerman, eds., Akademi, Berlin, 1988, pp. 114–125.

[40] B. KUMMER, *Newton's method based on generalized derivatives for nonsmooth functions: Convergence analysis*, in Advances in Optimization, Lecture Notes in Econom. and Math. Systems, W. Oettli and D. Pallaschke, eds., Springer, Berlin, 1992, pp. 171–194.

[41] J. LEE, Y. SUN, AND M. SAUNDERS, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim., 24 (2014), pp. 1420–1443.

[42] C. LEMARÉCHAL, *Numerical experiments in nonsmooth optimization*, in Progress in Nondifferentiable Optimization, E. A. Nurminski, ed., IIASA Press, Laxenburg, Austria, 1982, pp. 61–84.

[43] A. S. LEWIS AND M. L. OVERTON, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163.

[44] A. S. LEWIS AND S. ZHANG, *Nonsmoothness and a variable metric method*, J. Optim. Theory Appl., 165 (2015), pp. 151–171.

[45] J. L. MORALES AND J. NOCEDAL, *Remark on algorithm L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization*, ACM Trans. Math. Softw., 38 (2011), Article no. 7.

[46] J.-J. MOREAU, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C. R. Acad. Sci., 255 (1962), pp. 2897–2899.

[47] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate $O(1/k^2)$*, Soviet Math. Dokl., 27 (1983), pp. 372–376.

[48] Y. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Kluwer, Boston, 2004.

[49] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[50] P. OCHS, *Unifying Abstract Inexact Convergence Theorems for Descent Methods and Block Coordinate Variable Metric iPiano*, preprint, https://arxiv.org/abs/1602.07283, 2016.

[51] P. OCHS, J. FADILI, AND T. BROX, *Non-smooth non-convex bregman minimization: Unification and new algorithms*, J. Optim. Theory Appl., 181 (2018), pp. 244–278.

[52] P. OCHS AND T. POCK, *Adaptive Fista*, preprint, https://arxiv.org/abs/1711.04343, 2017.

[53] J.-S. PANG, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[54] P. PATRINOS, L. STELLA, AND A. BEMPORAD, *Forward-Backward Truncated Newton Methods for Convex Composite Optimization*, preprint, https://arxiv.org/abs/1402.6655, 2014.

[55] T. POCK AND A. CHAMBOLLE, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Curran Associates, Red Hook, NY, 2011, pp. 1762–1769.

[56] F. PORTA, M. PRATO, AND L. ZANNI, *A new steplength selection for scaled gradient methods with application to image deblurring*, J. Sci. Comput., 65 (2015), pp. 895–919.

[57] L. QI AND J. SUN, *A nonsmooth version of Newton's method*, Math. Program., 58 (1993), pp. 353–367.

[58] R. R. FLETCHER, *A limited memory steepest descent method*, Math. Program., 135 (2011), pp. 413–436.

[59] H. RAGUET, J. FADILI, AND G. PEYRÉ, *A generalized forward-backward splitting*, SIAM J. Imaging Sci., 6 (2013), pp. 1199–1226.

[60] S. M. ROBINSON, *Newton's method for a class of nonsmooth functions*, Set-Valued Anal., 2 (1994), pp. 291–305.

[61] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[62] S. Salzo, *The Variable Metric Forward-Backward Splitting Algorithm Under Mild Differentiability Assumptions*, preprint, https://arxiv.org/abs/1605.00952, 2016.

[63] M. Schmidt, *Graphical Model Structure Learning with ℓ_1-Regularization*, PhD thesis, University of British Columbia, Vancouver, BC, 2010.

[64] M. Schmidt, G. Fung, and R. Rosales, *Fast optimization methods for L1 regularization: A comparative study and two new approaches*, in Machine Learning: ECML 2007, Lecture Notes in Comput. Sci. 4701, Springer, Berlin, 2007, pp. 286–297.

[65] M. Schmidt, D. Kim, and S. Sra, *Projected Newton-type methods in machine learning*, in Optimization for Machine Learning, S. Sra, S. Nowozin, and S. Wright, eds., MIT Press, Cambridge, MA, 2011.

[66] 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 Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics, Proc. Mach. Learn. Res. 5, 2009, pp. 456–463; available at http://proceedings.mlr.press/v5/schmidt09a.html.

[67] L. Stella, A. Themelis, and P. Patrinos, *Forward-backward quasi-Newton methods for nonsmooth optimization problems*, Comput. Optim. Appl., 67 (2017), pp. 443–487.

[68] W. Su, S. Boyd, and R. Candes, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, in Adv. Neural Inf. Process. Syst. 27, Curran Associates, Red Hook, NY, 2014, pp. 2510–2518.

[69] J. F. Toland, *A duality principle for non-convex optimisation and the calculus of variations*, Arch. Ration. Mech. Anal., 71 (1979), pp. 41–61.

[70] M. Ulbrich, *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim., 13 (2002), pp. 805–841.

[71] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, MOS-SIAM Ser. Optim. 11, SIAM, Philadelphia, PA, 2011.

[72] L. van den Dries and C. Miller, *Geometric categories and o-minimal structures*, Duke Math. J., 84 (1996), pp. 497–540.

[73] B. C. Vũ, *A variable metric extension of the forward-backward-forward algorithm for monotone operators*, Numer. Funct. Anal. Optim., 34 (2013), pp. 1050–1065.

[74] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation*, SIAM J. Sci. Comput., 32 (2010), pp. 1832–1857.

[75] S. Wright, R. Nowak, and M. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493.

[76] J. Yu, S. V. N. Vishwanathan, S. Guenter, and N. Schraudolph, *A quasi-Newton approach to nonsmooth convex optimization problems in machine learning*, J. Mach. Learn. Res., 11 (2010), pp. 1145–1200.

[77] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Trans. Math. Softw., 23 (1997), pp. 550–560.