Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 20, No. 3, pp. 537–568, 2024

# EFFICIENT REGULARIZED PROXIMAL QUASI/NEWTON METHODS FOR LARGE-SCALE NONCONVEX COMPOSITE OPTIMIZATION PROBLEMS

Christian Kanzow and Theresa Lechner

*Dedicated to Masao Fukushima on the occasion of his 75th birthday.*

**Abstract:** Optimization problems with composite functions consist of an objective function which is the sum of a smooth and a (convex) nonsmooth term. This particular structure is exploited by the class of proximal gradient methods and some of their generalizations like proximal Newton and quasi-Newton methods. In this paper, we propose a regularized proximal quasi-Newton method whose main features are: (a) the method is globally convergent to stationary points, (b) the globalization is controlled by a regularization parameter, no line search is required, (c) the method can be implemented very efficiently based on a simple observation which combines recent ideas for the computation of quasi-Newton proximity operators and compact representations of limited-memory quasi-Newton updates. Numerical examples for the solution of convex and nonconvex composite optimization problems indicate that the method outperforms several existing methods.

**Key words:** *composite minimization, regularization, quadratic approximation, proximal quasi-Newton method, global convergence, limited memory methods, proximity operator, local error bound*

**Mathematics Subject Classification:** *49M15, 49M37, 65K05, 65K10, 90C06, 90C26, 90C30, 90C53*

## 1 Introduction

We consider the problem

$$\min_x \psi(x) := f(x) + \varphi(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $\varphi : \mathbb{R}^n \to \mathbb{R}$ is convex. In this formulation, the objective function $\psi$ is neither smooth nor convex, so a wide class of problems is covered, including problems in machine learning, compressed sensing, signal processing, and statistics. Although the assumption that $\varphi$ is real-valued (i.e., excludes the value $+\infty$) seems quite restrictive, the above formulation still comprises a considerably high number of applications in the above fields.

Probably the most prominent example in composite optimization, especially in the context of signal processing and compressed sensing, is the $\ell_1$-regularized least squares problem [4, 19, 32, 65], also called basic pursuit denoising, which tries to solve the problem

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1,$$


DOI: https://doi.org/10.61208/pjo-2023-036

where the quadratic term with $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$ has the purpose to find an approximate solution of $Ax \approx b$, whereas the $\ell_1$-term with a regularization parameter $\lambda > 0$ controls the sparsity of the solution. More details on this problem can be found in [19]. A wide class of more general applications combines this regularization $\varphi(x) = \lambda\|x\|_1$ with arbitrary convex [4, 11, 17, 27] or nonconvex [46] functions $f$ which are also covered by our setting. In particular, this includes problems with different loss functions like the logistic loss

$$f(x) := \tfrac{1}{m}\sum_{i=1}^{m} \log\big(1 + \exp(a_i^T x)\big),$$

see [10, 33, 36], or the nonconvex Student's $t$-loss

$$f(x) := \tfrac{1}{m}\sum_{i=1}^{m} \log\big(1 + (a_i^T x - b_i)^2\big),$$

for some data $a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$, cf. [1, 46]. These loss problems are typically used to classify data or reconstruct incomplete or blurred data. Alternative approaches are based on SVMs (support vector machines). The corresponding loss functions are usually combined with a suitable (frequently nonsmooth) regularization term which is added in order to impose some further properties like existence, stability or sparsity of solutions. For more applications of problem (1.1), we refer to [8, 17, 29] and references therein.

There are countless algorithms for determining solutions of composite optimization problems. These include semismooth Newton methods [39,46,49], interior point methods [32,33], trust-region methods [2,15], fixed point methods [13,14], or reformulations into a smooth problem with a forward backward envelope [60, 61], to name just a few. The focus in this paper, however, is on proximal-type methods, as these offer a very efficient way for solving many composite optimization problems.

Proximal-type methods for the solution of composite optimization problems trace back to the generalized proximal-point method by Fukushima and Mine [23]. The general purpose algorithm for solving (1.1) is to use a quadratic approximation of the smooth part $f$ and to solve, in each step, a problem of the form

$$\min_x f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T H_k(x - x^k) + \varphi(x), \tag{1.2}$$

where $x^k$ denotes the current iterate. A crucial point for developing such algorithms is the choice of the matrix $H_k \in \mathbb{R}^{n\times n}$.

First-order methods use $H_k$ as a positive multiple of the identity matrix and are often referred to as proximal gradient methods. In many cases, $H_k$ is constant over the complete algorithm and does not depend on the iteration. The main advantage of these algorithms is that the solution of the subproblems (1.2) can be done very efficiently or sometimes even analytically (depending on the function $\varphi$). A prominent method of this kind is the Iterative Shrinkage Threshold Algorithm [4] and its separable extension [62]. Many improvements are possible to accelerate this approach [4, 26, 51, 66].

Proximal quasi-Newton and variable metric proximal methods choose $H_k$ by using a suitable updating technique, hence $H_k$ changes from iteration to iteration, and the quadratic part in the subproblem (1.2) typically yields a much better approximation of the nonlinear function $f$ than for the simple choice in proximal gradient methods. On the other hand, this more advanced choice of $H_k$ makes the subproblem (1.2) more difficult to solve, in

particular, analytic solutions are usually no longer available. In order to deal with this disadvantage, suitable methods therefore allow to solve these subproblems only inexactly. Global convergence results for these proximal quasi-Newton methods are available in [7, 8, 21, 28, 41, 50, 60], which are based on different inexactness criteria, line search techniques, and appropriate assumptions regarding the choice of the sequence $\{H_k\}$ (usually uniform boundedeness and positive definiteness).

Using (at least approximate) second-order information in $H_k$ yields the class of proximal Newton methods [5, 6, 34, 35, 53, 54]. The standard technique to ensure global convergence is to combine the solution of the subproblems with some backtracking strategy. Similar to proximal quasi-Newton methods, these proximal Newton approaches often use different criteria to solve (1.2) only inexactly. Despite having suitable global convergence properties, they also inherit the local fast convergence known from Newton-type methods under certain assumptions, see [10, 24, 36, 47, 58, 67] for several realizations.

In this article, we present a different approach, in which $H_k$ is the sum of a matrix $B_k$ and a multiple $\mu_k I$ of the identity matrix for some regularization parameter $\mu_k > 0$. The purpose is to chose $B_k$ as a (limited memory) quasi-Newton approximation to the Hessian $\nabla^2 f(x^k)$ in the current iterate and to increase or decrease $\mu_k$ according to a trust-region-type framework, depending on the merit of the iteration. As a consequence, the method gets along without using a classical line search approach, which turns out to be more efficient in numerical examples. Moreover, and this is a central point of our contribution, if $B_k$ is chosen as a limited memory quasi-Newton approximation of $\nabla^2 f(x^k)$, we combine the theory of Becker et al. [6] with the compact representation of these limited memory quasi-Newton methods in order to get a very efficient solution technique for the resulting subproblems (1.2). To the authors' knowledge, there exist only few publications dealing with limited memory matrices and the advantages of their compact representation for proximal-type methods, e.g. [30, 34]. The combination with the results in [6] outline the benefits and makes this technique applicable to a wider class of applications, especially for large scale problems.

The idea of combining the regularization and (proximal) quasi-Newton techniques goes back to the corresponding methods for smooth problems ($\varphi = 0$), where the subproblem (1.2) reduces to $H_k(x - x^k) = -\nabla f(x^k)$, at least if $H_k$ is positive semidefinite. Some improvements [37, 59, 63, 64] have been made similar to our approach. Trust-region methods for nonsmooth problems in the form of (1.1) are also considered in different papers [15, 20, 31, 56]. Techniques for the regularization of proximal quasi-Newton methods are investigated in several variations in literature. The proximal Newton method by Lee, Sun, Saunders [36] does not explicitly use a regularization parameter, but the application to proximal quasi-Newton methods covers this idea if the regularization parameter tends to zero. A similar approach is used in the authors' work in [29]. Regularization of $B_k$ by adding a positive multiple of the identity matrix is also used in [24, 58], but convergence is only shown for convex functions $f$. Approaches for solving the subproblems inexactly are investigated in [35, 67]. Finally, we mention that the essence of the proximal quasi-Newton method from Karimi and Vavasis [30] is similar to our approach. However, they only consider $\ell_1$-regularized least squares problems and allow $H_k$ to be a 'diagonal minus rank-1'-matrix. Furthermore, they do not use a regularization of $H_k$. Their theoretical approach is generalized by the work of Becker et al. [6]. We outline the main differences of these methods to the current one after stating our algorithm in Section 3.

The paper is organized as follows. We first give an overview of some background material in Section 2. The regularized proximal quasi-Newton method itself is presented in Section 3. Global convergence of this method is shown in Section 4 under fairly mild assumptions in the trust-region framework. In addition, under an error bound assumption we prove that

a sequence generated by our method is convergent and summable. Section 6 describes the new trick for an efficient solution of the resulting subproblems (1.2) if $B_k$ is computed by a limited memory quasi-Newton technique. Numerical results and comparisons with some standard solvers are provided in Section 7 with a focus on proximal-type methods. We conclude with some final remarks in Section 8.

Notation: The set of all symmetric positive definite matrices in $\mathbb{R}^{n\times n}$ is denoted by $\mathbb{S}^n_{++}$. We write $A \succeq B$ or $A \succ B$, if the matrix $A - B$ is positive semidefinite or positive definite, resp. For a symmetric matrix $H \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the smallest and largest eigenvalue of $H$, respectively. Furthermore, $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ are the Euclidean norm and scalar product, while $\|\cdot\|_H$ and $\langle\cdot,\cdot\rangle_H$ denote the norm and scalar product with respect to $H \in \mathbb{S}^n_{++}$, i.e. $\langle x, y\rangle_H = x^T Hy$ and $\|x\|_H = \sqrt{\langle x, x\rangle_H}$. We write $x_{\mathcal{I}}$ to describe the subvector of $x \in \mathbb{R}^n$ consisting of all entries $x_i$ with $i \in \mathcal{I}$. Finally, for a sequence $\{x^k\}$, we write $x^k \to_K \bar{x}$ for some infinite subset $K \subseteq \mathbb{N}$ if the subsequence $\{x^k\}_K$ converges to $\bar{x}$.

## 2 Preliminaries

This section summarizes some background material and states a preliminary result which will be used in order to derive and investigate our regularized proximal quasi-Newton method.

The *subdifferential* $\partial\varphi(x)$ of a convex function $\varphi : \mathbb{R}^n \to \mathbb{R}$ in a point $x \in \mathbb{R}^n$ is defined as

$$\partial\varphi(x) := \big\{s \in \mathbb{R}^n \mid \varphi(y) \geq \varphi(x) + s^T(y - x)\ \forall y \in \mathbb{R}^n\big\}.$$

Some properties of this subdifferential are summarized in the following proposition, cf. the classical monograph [57] by Rockafellar for more details.

**Proposition 2.1.** *Let $\varphi : \mathbb{R}^n \to \mathbb{R}$ be convex. Then the following statements hold:*

1. *$\partial\varphi(x) \neq \emptyset$ for every $x \in \mathbb{R}^n$ [57, Theorem 23.4].*
2. *$\partial\varphi$ maps bounded sets onto bounded sets [57, Theorem 24.7].*
3. *Let $\{x^k\}, \{s^k\} \subset \mathbb{R}^n$ be sequences such that $x^k \to x^*$, $s^k \to s^*$ and $s^k \in \partial\varphi(x^k)$ for all $k \in \mathbb{N}$. Then $s^* \in \partial\varphi(x^*)$ (closedness of the subdifferential) [57, Theorem 24.4].*
4. *$x^* \in \arg\min\varphi$ if and only if $0 \in \partial\varphi(x^*)$ (Fermat's rule) [3, Theorem 16.3].*

Note that, in general, parts (a) and (b) do not hold if $\varphi$ is extended-valued.

The basis of proximal-type methods is the proximity operator, introduced by Moreau [48]. For a convex function $\varphi : \mathbb{R}^n \to \mathbb{R}$ and a positive definite matrix $H \in \mathbb{S}^n_{++}$, the *proximity operator* with respect to $H$ is the mapping

$$x \mapsto \operatorname{prox}^H_\varphi(x) := \arg\min_y \Big\{\varphi(y) + \frac{1}{2}(y - x)^T H(y - x)\Big\}.$$

Since $H$ is positive definite, the regularization $\varphi(y) + \frac{1}{2}(y-x)^T H(y-x)$ is strongly convex. Hence, it has a unique minimizer for every $x \in \mathbb{R}^n$, thus the proximity operator is well-defined. If $H$ is the identity matrix, we simply write

$$\operatorname{prox}_\varphi(x) := \operatorname{prox}^I_\varphi(x).$$

Some basic properties of the proximity operator are summarized in the following result.

**Proposition 2.2.** *Let $\varphi : \mathbb{R}^n \to \mathbb{R}$ be convex and $H \in \mathbb{S}^n_{++}$. Then the following statements hold:*

1. *The proximity operator is firmly nonexpansive with respect to the norm induced by $H$ [45, Lemma 3.1.1], i.e. for any $x, y \in \mathbb{R}^n$ there holds*

$$\left\| \operatorname{prox}_\varphi^H(x) - \operatorname{prox}_\varphi^H(y) \right\|_H^2 \leq \left\langle \operatorname{prox}_\varphi^H(x) - \operatorname{prox}_\varphi^H(y), x - y \right\rangle_H.$$

2. *$p = \operatorname{prox}_\varphi^H(x)$ if and only if $p \in x - H^{-1}\partial\varphi(p)$ [3, Proposition 16.44].*

Let $x, d \in \mathbb{R}^n$. Then, the directional derivative of $\psi$ in $x$ and direction $d$ is the one-sided limit

$$\psi'(x; d) := \lim_{t \downarrow 0} \frac{\psi(x + td) - \psi(x)}{t}.$$

We call $x^* \in \mathbb{R}^n$ a *stationary point* of $\psi$ or a *stationary point* of problem (1.1) if $0 \in \nabla f(x^*) + \partial\varphi(x^*)$. Thus, we obtain the following characterizations:

$$\begin{aligned} x^* \text{ stationary point of } \psi \quad &\iff -\nabla f(x^*) \in \partial\varphi(x^*) \\ &\iff \psi'(x^*; d) \geq 0 \text{ for all } d \in \mathbb{R}^n \\ &\iff x^* = \operatorname{prox}_\varphi^H(x^* - H^{-1}\nabla f(x^*)), \end{aligned} \tag{2.1}$$

where the second line follows from [3, Proposition 17.14] and the final one is a consequence of Proposition 2.2(b), which is independent of the particular matrix $H \in \mathbb{S}_{++}^n$. Given $x \in \mathbb{R}^n$ and $H \in \mathbb{S}_{++}^n$, it follows that the norm of the corresponding residual

$$r_H(x) := \arg\min_d \left\{ \nabla f(x)^T d + \frac{1}{2} d^T H d + \varphi(x + d) \right\} = \operatorname{prox}_\varphi^H\left(x - H^{-1}\nabla f(x)\right) - x$$

can be used to measure the stationarity of $x$. For the special case $H = I$, we again simplify the notation and write

$$r(x) := r_I(x).$$

The relation between $\|r_H(x)\|$ and $\|r_{\tilde{H}}(x)\|$ for two different matrices $H, \tilde{H}$ is stated in the next result.

**Lemma 2.3.** *Let $x \in \mathbb{R}^n$ and $H, \tilde{H} \in \mathbb{S}_{++}^n$. Then*

$$\|r_{\tilde{H}}(x)\| \leq \left(1 + \frac{\lambda_{\max}(\tilde{H})}{\lambda_{\min}(H)}\right) \cdot \frac{\lambda_{\max}(H)}{\lambda_{\min}(\tilde{H})} \cdot \|r_H(x)\|.$$

*Proof.* By [62, Lemma 3], we get

$$\|r_{\tilde{H}}(x)\| \leq \frac{1 + \lambda_{\max}(Q) + \sqrt{1 - 2\lambda_{\min}(Q) + \lambda_{\max}(Q)^2}}{2} \, \frac{\lambda_{\max}(H)}{\lambda_{\min}(\tilde{H})} \cdot \|r_H(x)\|,$$

where $Q := H^{-1/2}\tilde{H}H^{-1/2}$ is also positive definite. The claim follows from the inequalities

$$1 - 2\lambda_{\min}(Q) + \lambda_{\max}(Q)^2 \leq 1 + \lambda_{\max}(Q)^2 \leq (1 + \lambda_{\max}(Q))^2$$

and $\lambda_{\max}(Q) \leq \lambda_{\max}(\tilde{H})/\lambda_{\min}(H)$. The latter estimate follows from

$$\begin{aligned} \lambda_{\max}(Q) &= \max_{x \neq 0} \frac{x^T H^{-1/2}\tilde{H}H^{-1/2}x}{x^T x} = \max_{z \neq 0} \frac{z^T \tilde{H} z}{z^T H z} = \max_{z \neq 0} \left( \frac{z^T \tilde{H} z}{z^T z} \frac{z^T z}{z^T H z} \right) \\ &\leq \left( \max_{z \neq 0} \frac{z^T \tilde{H} z}{z^T z} \right) \left( \max_{z \neq 0} \frac{1}{\frac{z^T H z}{z^T z}} \right) = \lambda_{\max}(\tilde{H}) \frac{1}{\min_{z \neq 0} \frac{z^T H z}{z^T z}} = \lambda_{\max}(\tilde{H}) \cdot \frac{1}{\lambda_{\min}(H)}, \end{aligned}$$

and this completes the proof. □

## 3 The Regularized Proximal Quasi-Newton Method

This section contains a detailed derivation and discussion of our regularized proximal quasi-Newton method. Given an iterate $x^k \in \mathbb{R}^n$, consider the subproblem

$$\min_d q_k(d) \quad \text{with} \quad q_k(d) := f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T B_k d + \varphi(x^k + d), \tag{3.1}$$

where the first part is a quadratic approximation to the smooth function $f$, with $B_k$ being a (possibly bad) approximation of the (possibly not existing) Hessian $\nabla^2 f(x^k)$. The main idea of proximal quasi-Newton methods is then to compute $d^k$ as a solution of the subproblem (3.1), and to set $x^{k+1} := x^k + d^k$ provided that $d^k$ is accepted by a suitable line search or trust-region strategy in order to obtain global convergence results. Here, the globalization is done by a regularization parameter, no line search is required (which might result in many function evaluations), and no trust-region radius is needed (in particular, no trust-region-type subproblem has to be solved). Instead, however, additional evaluations of the proximity operator may be required, which can be quite expensive. Nevertheless, numerical tests show that this additional effort leads to significantly fewer iterations and thus lower overall costs, and, furthermore, trust-region methods are more appropriate, especially for non-convex global optimization problems.

The regularized proximal quasi-Newton method therefore considers the regularized approximation

$$\hat{q}_k(d) := q_k(d) + \tfrac{1}{2}\mu_k \|d\|^2 = f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T (B_k + \mu_k I) d + \varphi(x^k + d) \tag{3.2}$$

with some parameter $\mu_k > 0$. To control the success of a candidate $d^k$, which is a solution of the regularized subproblem $\min_d \hat{q}_k(d)$, we define the *predicted reduction* of $\psi$ as

$$\text{pred}_k := \psi(x^k) - q_k(d^k) = -\big(\nabla f(x^k)^T d^k + \varphi(x^k + d^k) - \varphi(x^k)\big) - \tfrac{1}{2}(d^k)^T B_k d^k$$

and the *actual reduction* of $\psi$ as $\text{ared}_k := \psi(x^k) - \psi(x^k + d^k)$. The ratio $\rho_k := \text{ared}_k \,/\, \text{pred}_k$ between these quantities is, similar to trust-region methods [18], used to control the update of the regularization parameter and the iterate. Since $B_k$ does not need to be positive definite, we have to take into account that a minimizer of $\hat{q}_k$ may not exist or the corresponding value $\text{pred}_k$ is not (sufficiently) positive. These situations are handled as unsuccessful steps. Altogether, this motivates the following algorithm.

**Algorithm 3.1** (Regularized Proximal Quasi-Newton Method)**.**

(S.0) Choose $x^0 \in \mathbb{R}^n$, parameters $\mu_0 > 0$, $p_{\min} \in (0, \frac{1}{2})$, $c_1 \in (0, \frac{1}{2})$, $c_2 \in (c_1, 1)$, $\sigma_1 \in (0,1), \sigma_2 > 1$, and set $k := 0$.

(S.1) If $x^k$ satisfies a suitable termination criterion: STOP.

(S.2) Choose $B_k \in \mathbb{R}^{n \times n}$, and find a solution $d^k$ of the problem

$$\min_d \hat{q}_k(d) = f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T (B_k + \mu_k I) d + \varphi(x^k + d). \tag{3.3}$$

If this problem has no solution, or if

$$\text{pred}_k \le p_{\min} \|d^k\| \cdot \|r(x^k)\|, \tag{3.4}$$

set $x^{k+1} := x^k$, $\mu_{k+1} := \sigma_2 \mu_k$, and go to (S.4). Otherwise go to (S.3).

(S.3) Set $\rho_k := \text{ared}_k \,/\, \text{pred}_k$ and perform the following updates:

$$x^{k+1} := \begin{cases} x^k & \text{if } \rho_k \le c_1, \\ x^k + d^k & \text{otherwise,} \end{cases} \qquad \mu_{k+1} := \begin{cases} \sigma_2 \mu_k & \text{if } \rho_k \le c_1, \\ \mu_k & \text{if } c_1 < \rho_k \le c_2, \\ \sigma_1 \mu_k & \text{otherwise.} \end{cases}$$

(S.4) Update $k \leftarrow k + 1$, and go to (S.1).

In the following, we call an iteration $k$

- *unsuccessful*, if (S.3) is skipped or $\rho_k \le c_1$,
- *successful,* if $c_1 < \rho_k \le c_2$,
- *highly successful,* if $\rho_k > c_2$.

Note that, in an unsuccessful iteration, both (S.2) and (S.3) keep the current iterate $x^k$ and choose a larger regularization parameter. In all other iterations, we update $x^{k+1}$ and either keep the regularization parameter $\mu_k$ (in all successful iterations) or reduce this parameter (in all highly successful iterations). We also stress that a test like (3.4) is not required by trust-region methods since, there, the corresponding predicted reduction is automatically positive, whereas this cannnot be guaranteed in our setting. Whenever we reach (S.3), however, the value of $\text{pred}_k$ is (sufficiently) positive, which, in turn, implies that the overall method is well-defined.

We briefly discuss the differences between Algorithm 3.1 and some affiliated methods. The methods in [24, 58] are based on a similar regularization than ours, where the regularization parameter is only increased if a suitable criterion is not satisfied for the solution of the subproblems. In contrast to our method, they do not consider the possibility to reduce the regularization parameter if an iterate is highly successful. Convergence is shown under the assumption of strong convexity of $f$. Furthermore, they combine the method with an inexactness criterion on the subproblem and use a FISTA-type acceleration. In this case, a main assumption on $f$ is convexity.

The method by Karimi and Vavasis [30] is a basic proximal Newton method for solving $\ell_1$-regularized least squares problems. No regularization is included and their analysis focusses on $H_k$ being a rank-1 modification of a multiple of the identity.

The inexact algorithms by Lee and Wright [35] use two different types of regularization: $H_k = B_k + \mu_k I$ or $H_k = \mu_k B_k$ with a positive regularization parameter $\mu_k$, which is initially set to 1 in each step and increased until a sufficient decrease condition is satisfied. In contrast to our method, it is not possible to choose $\mu_k$ small when the iterate is close to a solution. Convergence is shown for $\nabla f$ being Lipschitz continuous (but $f$ is not necessarily convex). Moreover, some improved convergence results are provided for strongly convex functions.

Yue et al. [67] develop another inexact regularized proximal Newton method. A main difference to our approach is that, instead of an approximation $B_k$, the exact Hessian of $f$ is used and the regularization parameter $\mu_k$ is chosen due to the optimality of the current iterate, and not based on the quality of the current update. Furthermore, the subproblems are solved inexactly, and an Armijo-type line search is performed. The convergence proof needs convexity of $f$ and uses an error bound.

In contrast to these methods, we do not provide a theory for inexact solutions of the subproblems in (S.2). It turns out that this is not necessary since these problems can be solved very efficiently and with high accuracy in our numerical examples.

In view of (2.1), we know that $x^k$ is a stationary point of $\psi$ if and only if $r(x^k) = 0$. Combining this property with the (uniform) continuity of $r(\cdot)$ yields an appropriate termination criterion for Algorithm 3.1. For the method to be well-defined, we need a similar property for the value $d^k$ (note that, by definition, we have $d^k = r_{B_k+\mu_k I}(x^k)$, if the matrix $B_k + \mu_k I$ is positive definite).

**Lemma 3.2.** *If $d^k = 0$ in Algorithm 3.1, then $x^k$ is a stationary point of $\psi$. The converse is true if $B_k + \mu_k I$ is positive definite.*

*Proof.* Assume that $d^k = 0$. From the definition of $d^k$ and Fermat's rule, we get

$$0 \in \nabla f(x^k) + (B_k + \mu_k I)d^k + \partial\varphi(x^k + d^k).$$

Plugging in $d^k = 0$ yields $0 \in \nabla f(x^k) + \partial\varphi(x^k)$, which is the desired result. Conversely, let $B_k + \mu_k I$ be positive definite and $x^k$ a stationary point of $\psi$. Then $-\nabla f(x^k) \in \partial\varphi(x^k)$, which yields $\varphi(x^k + d) \geq \varphi(x^k) - \nabla f(x^k)^T d$ for every $d \in \mathbb{R}^n$. Thus,

$$\begin{aligned}\hat{q}_k(0) &= f(x^k) + \varphi(x^k) \leq f(x^k) + \nabla f(x^k)^T d + \varphi(x^k + d)\\ &\leq f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T (B_k + \mu_k I)d + \varphi(x^k + d) = \hat{q}_k(d)\end{aligned}$$

for all $d \in \mathbb{R}^n$. Hence, $d^k = 0$ due to the uniqueness of the global minimum for $B_k + \mu_k I$ being positive definite. □

It is not difficult to see that the converse statement in Lemma 3.2 may not hold if $B_k + \mu_k I$ is only positive semidefinite or indefinite. Hence, the termination criterion in (S.1) of Algorithm 3.1 should rely on $r(x^k)$ instead of $d^k$ as positive definiteness of $B_k + \mu_k I$ is not required.

## 4 Global Convergence Theory

In this section, we investigate the global convergence properties of Algorithm 3.1. Similar to convergence results for trust-region methods this means that $\liminf_{k\to\infty} \|r(x^k)\| = 0$ or $\lim_{k\to\infty} \|r(x^k)\| = 0$, depending on the assumptions. Using (2.1), this implies that every accumulation point is a stationary point of $\psi$. To prove this, we assume that Algorithm 3.1 generates an infinite sequence $\{x^k\}$. Though, formally, we did not specify the termination criterion in (S.1), any suitable stopping criterion will include a test whether the current point $x^k$ is already a stationary point of the given optimization problem. Now, to simplify some of the subsequent phrases, we therefore assume throughout this section that none of the iterations $x^k$ is already a stationary point. Then, by Lemma 3.2, we have $d^k \neq 0$ for all $k$.

The subsequent global convergence analysis of Algorithm 3.1 does not require the matrices $B_k$ to be good approximations of the corresponding (possibly not existing) Hessians $\nabla^2 f(x^k)$. We only need that the sequence $\{B_k\}$ is bounded. Before presenting the two main global convergence theorems, we establish some technical results.

**Lemma 4.1.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices. Assume that $\mu_k \to \infty$ and $\{x^k\} \subset \mathbb{R}^n$ converges to a nonstationary point $\overline{x}$ of $\psi$. Then*

$$\limsup_{k\to\infty} \frac{\|r(x^k)\|}{\|r_{B_k+\mu_k I}(x^k)\| \cdot \mu_k} \leq 1.$$

*Proof.* The assumptions imply that $B_k + \mu_k I$ is positive definite for all sufficiently large $k$. Furthermore, $\|r_{B_k+\mu_k I}(x^k)\| \neq 0$ for sufficiently large $k \geq 0$ since $\overline{x}$ is not a stationary point of $\psi$ and $r$ is continuous. Thus, we can apply Lemma 2.3 with $H = B_k + \mu_k I$ and $\tilde{H} = I$ to get

$$\frac{\|r(x^k)\|}{\|r_{B_k+\mu_k I}(x^k)\|} \leq \Big(1 + \frac{1}{\lambda_{\min}(B_k) + \mu_k}\Big) \cdot \big(\lambda_{\max}(B_k) + \mu_k\big).$$

Dividing this estimate by $\mu_k$, using the boundedness of the sequence $\{B_k\}$, and taking $k \to \infty$, it follows that the expression on the right-hand side tends to 1, which yields the claim. □

Recall that if $B_k + \mu_k I$ is positive definite, step $d^k$ can be written as $d^k = r_{B_k+\mu_k I}(x^k)$. In the next result, we show that this sequence is a vanishing sequence under the assumptions that the sequence $\{\mu_k\}$ tends to $+\infty$ and $\{x^k\}$ is bounded.

**Proposition 4.2.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices. Assume that $\mu_k \to \infty$ and the sequence $\{x^k\} \subset \mathbb{R}^n$ generated by Algorithm 3.1 is bounded. Let $d^k := r_{B_k+\mu_k I}(x^k)$. Then $d^k \to 0$.*

*Proof.* Note that the boundedness of the sequence $\{B_k\}$ and $\mu_k \to \infty$ imply that $d^k$ is well defined for sufficiently large $k$. Moreover, the definition of successful steps implies that the sequence $\{\psi(x^k)\}$ is a monotonically decreasing. Hence, for all $k \in \mathbb{N}$ sufficiently large, we have

$$\begin{aligned}
\psi(x^0) \geq \psi(x^k) &= \hat{q}_k(0) \geq \hat{q}_k(d^k) \\
&= f(x^k) + \nabla f(x^k)^T d^k + \frac{1}{2}(d^k)^T(B_k + \mu_k I)d^k + \varphi(x^k + d^k) \\
&\geq f(x^k) + \nabla f(x^k)^T d^k + \frac{1}{2}(d^k)^T(B_k + \mu_k I)d^k + \varphi(x^k) + (u^k)^T d^k
\end{aligned}$$

for some $u^k \in \partial\varphi(x^k)$. Since, by assumption, the sequences $\{x^k\}$ and $\{B_k\}$ are bounded and, therefore, the sequences $\{f(x^k)\}$, $\{\varphi(x^k)\}$, $\{\nabla f(x^k)\}$, and $\{u^k\}$ are bounded by the continuity of $f$, $\varphi$ and $\nabla f$ and Propositon 2.1 (a), the limiting behaviour of the right-hand side is dominated by the quadratic term $\frac{1}{2}(d^k)^T(B_k + \mu_k I)d^k$. Thus, this term is bounded from above, and the assumption $\mu_k \to \infty$ immediately implies $d^k \to 0$. □

The following result will be applied to the situation where we have only finitely many successful iterations, i.e., where $x^k$ stays constant eventually, say $x^k = x^{k_0}$ for all $k \geq k_0$ and some sufficiently large index $k_0 \in \mathbb{N}$. We formulate this result in a slightly more general context and assume that we have a nonstationary limit point $\overline{x}$. To avoid any ambiguity in the notation, we write $\bar{d}^k := r_{B_k+\mu_k I}(\overline{x})$, although, in the subsequent application, we will eventually have $\bar{d}^k = d^k$ since $\overline{x}$ corresponds to $x^{k_0}$ $(= x^k$ for all $k \geq k_0)$.

**Lemma 4.3.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices. Assume that $\mu_k \to \infty$ and $\overline{x}$ is a nonstationary point of $\psi$. Define $\bar{d}^k := r_{B_k+\mu_k I}(\overline{x})$, and let $s$ be an accumulation point of the sequence $\{\bar{d}^k/\|\bar{d}^k\|\}$. Then $\psi'(\overline{x}; s) < 0$.*

*Proof.* Using the previous result, we get $\bar{d}^k \to 0$. Furthermore, using Fermat's rule, we obtain

$$0 = \nabla f(\overline{x}) + (B_k + \mu_k I)\bar{d}^k + u^k \tag{4.1}$$

for some $u^k \in \partial\varphi(\overline{x} + \bar{d}^k)$. The boundedness of the subdifferential (Proposition 2.1 (b)) yields that the sequence $\{u^k\}$ is bounded. Thus, we can choose a subsequence $K \subset \mathbb{N}$ such that

$$\frac{\bar{d}^k}{\|\bar{d}^k\|} \to_K s \qquad \text{and} \qquad u^k \to_K \overline{u}.$$

The closedness of the subdifferential (Proposition 2.1 (c)) yields $\overline{u} \in \partial\varphi(\overline{x})$. By assumption, we therefore have $\nabla f(\overline{x}) + \overline{u} \neq 0$.

Furthermore, using the results of [36, Proposition 2.4], see also equation (2.16) in that paper, we obtain

$$\psi'(\overline{x}, \bar{d}^k) \leq -(\bar{d}^k)^T (B_k + \mu_k I)\bar{d}^k \leq -\big(\lambda_{\min}(B_k) + \mu_k\big)\|\bar{d}^k\|^2.$$

Since (4.1) implies $\|\nabla f(\overline{x}) + u^k\| = \|(B_k + \mu_k I)\bar{d}^k\| \leq (\|B_k\| + \mu_k)\|\bar{d}^k\|$, we get

$$\psi'(\overline{x}, \bar{d}^k) \leq -\big(\lambda_{\min}(B_k) + \mu_k\big)\|\bar{d}^k\|^2 \leq -\|\nabla f(\overline{x}) + u^k\| \cdot \frac{\lambda_{\min}(B_k) + \mu_k}{\|B_k\| + \mu_k} \cdot \|\bar{d}^k\|.$$

Thus, the sublinearity of $\psi'(\overline{x}, \cdot)$ yields

$$\psi'\big(\overline{x}, \frac{\bar{d}^k}{\|\bar{d}^k\|}\big) \leq -\|\nabla f(\overline{x}) + u^k\| \cdot \frac{\lambda_{\min}(B_k) + \mu_k}{\|B_k\| + \mu_k}.$$

For $k \to_K \infty$, the right-hand side converges to $-\|\nabla f(\overline{x}) + \overline{u}\|$. Since $\varphi$ is real-valued, the directional derivative $\psi'(\overline{x}, \cdot)$ is continuous, and we obtain

$$\psi'(\overline{x}, s) = \lim_{K \ni k \to \infty} \psi'\Big(\overline{x}, \frac{\bar{d}^k}{\|\bar{d}^k\|}\Big) \leq -\|\nabla f(\overline{x}) + u\| < 0.$$

This completes the proof. □

We now apply the previous result to show that there always exist infinitely many successful or highly successful iterations.

**Lemma 4.4.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices. Then Algorithm 3.1 performs infinitely many successful or highly successful steps.*

*Proof.* We follow the proof of [59] and assume, by contradiction, that there exists $k_0 \in \mathbb{N}$ such that all steps $k \geq k_0$ are unsuccessful. This implies $x^k = x^{k_0}$ for all $k \geq k_0$ and, due to the implicit assumption that Algorithm 3.1 generates an infinite sequence, that $\mu_k \to +\infty$. Since $\{B_k\}$ is a bounded sequence, the matrices $B_k + \mu_k I$ are therefore positive definite for all sufficiently large $k$. In view of Lemma 3.2 and $d^k \neq 0$ (otherwise we would have stopped after finitely many iterations), it follows that $x^{k_0}$ is a nonstationary point of $\psi$, i.e., $r(x^{k_0}) \neq 0$. Moreover, the positive definiteness of $B_k + \mu_k I$ also guarantees that the search directions $d^k$ are well-defined. In view of Lemma 4.1, we have

$$\frac{\|r(x^k)\|}{\|d^k\|\mu_k} < \frac{1}{2p_{\min}}$$

for sufficiently large $k$ (recall that $p_{\min} < \frac{1}{2}$ and $d^k = r_{B_k + \mu_k I}(x^k)$). Using $\hat{q}_k(d^k) \leq \hat{q}_k(0)$, we then obtain

$$\text{pred}_k = \psi(x^k) - q_k(d^k) = \psi(x^k) - \hat{q}_k(d^k) + \frac{\mu_k}{2}\|d^k\|^2$$

$$\geq \psi(x^k) - \hat{q}_k(0) + \frac{\mu_k}{2}\|d^k\|^2 = \frac{\mu_k}{2}\|d^k\|^2 > p_{\min}\|r(x^k)\| \cdot \|d^k\|. \tag{4.2}$$

Hence, for all sufficiently large $k$, Algorithm 3.1 performs (S.3). Since all iterations $k \geq k_0$ are unsuccessful, this means $\text{ared}_k \leq c_1 \,\text{pred}_k$. It follows that

$$\psi(x^{k_0} + d^k) - \psi(x^{k_0}) \geq c_1\big(\nabla f(x^{k_0})^T d^k + \varphi(x^{k_0} + d^k) - \varphi(x^{k_0}) + \tfrac{1}{2}(d^k)^T B_k d^k\big).$$

Setting $t_k = \|d^k\|$ and dividing this estimate by $t_k$ yields

$$\frac{\psi(x^{k_0} + t_k \frac{d^k}{\|d^k\|}) - \psi(x^{k_0})}{t_k}$$
$$\geq c_1\left(\nabla f(x^{k_0})^T \frac{d^k}{\|d^k\|} + \frac{\varphi(x^{k_0} + t_k \frac{d^k}{\|d^k\|}) - \varphi(x^{k_0})}{t_k} + \frac{1}{2}\frac{(d^k)^T}{\|d^k\|} B_k d^k\right).$$

Choosing a subsequence $K$ such that $d^k/\|d^k\| \to s$, and using the local Lipschitz continuity of $\psi$, the left-hand side converges to the directional derivative $\psi'(x^{k_0}; s)$ when taking the limit in $K$. In the same way, the limit of the second term on the right-hand side converges to $\varphi'(x^{k_0}; s)$. Thus, using $d^k \to 0$, see Proposition 4.2, and the boundedness of $\{B_k\}$, taking the limit on $K$ in the entire estimate gives $\psi'(x^{k_0}; s) \geq c_1 \psi'(x^{k_0}; s)$. Since $c_1 \in (0, 1)$, this yields $\psi'(x^{k_0}; s) \geq 0$, a contradiction to Lemma 4.3. This shows that there are infinitely many successful or highly successful iterations. □

We next formulate two global convergence results. The corresponding statements are similar to those known for trust-region methods in, e.g., unconstrained optimization.

**Theorem 4.5.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices, and assume that $\psi$ is bounded from below. Then any sequence $\{x^k\}$ generated by the regularized proximal Newton-type method (Algorithm 3.1) satisfies* $\liminf_{k\to\infty} \|r(x^k)\| = 0$.

*Proof.* Let $\mathcal{S} \subset \mathbb{N}$ be the (infinite) set of successful or highly successful iterations. Contrary to the claim, assume that $\liminf_{k\to\infty} \|r(x^k)\| > 0$. Then there exists $k_0 \in \mathbb{N}$ and $\varepsilon > 0$ such that $\|r(x^k)\| \geq \varepsilon$ for all $k \geq k_0$. By the definition of successful steps, we get

$$\psi(x^k) - \psi(x^{k+1}) \geq c_1 \,\text{pred}_k \geq p_{\min} c_1 \|d^k\| \cdot \|r(x^k)\| \geq p_{\min} c_1 \varepsilon \|d^k\|$$

for all $k \in \mathcal{S}, k \geq k_0$. Since $\psi$ is bounded from below, summation yields

$$\infty > \sum_{k=0}^{\infty} \big[\psi(x^k) - \psi(x^{k+1})\big] = \sum_{k\in\mathcal{S}} \big[\psi(x^k) - \psi(x^k + d^k)\big] \geq p_{\min} c_1 \varepsilon \sum_{k\in\mathcal{S}} \|d^k\|.$$

Taking into account that $x^k$ is not updated in unsuccessful steps, it follows that

$$\infty > \sum_{k\in\mathcal{S}} \|d^k\| = \sum_{k\in\mathcal{S}} \|x^{k+1} - x^k\| = \sum_{k=0}^{\infty} \|x^{k+1} - x^k\|.$$

Hence, $\{x^k\}$ is a Cauchy sequence and therefore convergent to some $\overline{x} \in \mathbb{R}^n$. Since $\|r(\overline{x})\| = \lim_{k\to\infty} \|r(x^k)\| \geq \varepsilon$, $\overline{x}$ is not a stationary point of $\psi$.

By Lemma 4.4, there are infinitely many successful or highly successful steps and, as shown above, we have $\|d^k\| \to_{\mathcal{S}} 0$. Similar to (4.1) there holds

$$0 = \nabla f(x^k) + (B_k + \mu_k I)d^k + u^k$$

for some $u^k \in \partial\varphi(x^k + d^k)$. Assuming that $\{\mu_k\}_{\mathcal{S}}$ is bounded, $(B_k + \mu_k I)d^k$ converges to 0 for $k \to_{\mathcal{S}} \infty$. Furthermore, Proposition 2.1 (b), (c) yields that $\{u^k\}_{\mathcal{S}}$ is bounded and we can choose a subsequence $K \subset \mathcal{S}$ such that $u^k \to_K \overline{u}$ with $\overline{u} \in \partial\varphi(\overline{x})$. Taking the limit $K \ni k \to \infty$ in the above equation then yields $0 = \nabla f(\overline{x}) + \overline{u} \in \nabla f(\overline{x}) + \partial\varphi(\overline{x})$, in contradiction to the nonstationarity of $\overline{x}$.

Hence, without loss of generality, we have $\{\mu_k\}_{\mathcal{S}} \to \infty$. It follows that $\{\mu_k\} \to \infty$ since $\mu_k$ cannot decrease during unsuccessful iterations. This implies that Algorithm 3.1 also performs infinitely many unsuccessful iterations. On the other hand, in the same way as (4.2), we get

$$\operatorname{pred}_k \geq p_{\min}\|d^k\| \cdot \|r(x^k)\| \geq p_{\min}\varepsilon\|d^k\|$$

for sufficiently large $k$. For every such $k$, there exists $\xi^k$ on the straight line between $x^k$ and $x^k + d^k$ such that $f(x^k + d^k) - f(x^k) = \nabla f(\xi^k)^T d^k$. By the convergence of $\{x^k\}$ to $\overline{x}$ and since $\{d^k\} \to 0$ in view of Proposition 4.2, the sequence $\{\xi^k\}$ also converges to $\overline{x}$. Thus, we obtain

$$\begin{aligned}
\big|\rho_k - 1\big| &= \Big|\frac{\operatorname{ared}_k}{\operatorname{pred}_k} - 1\Big| = \Big|\frac{\psi(x^k) - \psi(x^k + d^k)}{\psi(x^k) - q_k(d^k)} - 1\Big| \\
&= \Big|\frac{\psi(x^k + d^k) - q_k(d^k)}{\psi(x^k) - q_k(d^k)}\Big| \\
&\leq \frac{1}{p_{\min}\varepsilon}\,\frac{\big|f(x^k + d^k) - f(x^k) - \nabla f(x^k)^T d^k\big| + \frac{1}{2}\big|(d^k)^T B_k d^k\big|}{\|d^k\|} \\
&\leq \frac{1}{p_{\min}\varepsilon}\,\frac{\big|\nabla f(\xi^k)^T d^k - \nabla f(x^k)^T d^k\big|}{\|d^k\|} + \frac{1}{2p_{\min}\varepsilon}\Big|(d^k)^T B_k \frac{d^k}{\|d^k\|}\Big| \longrightarrow 0
\end{aligned}$$

for $k \to \infty$. Hence, $\{\rho_k\} \to 1$, i.e., eventually all steps are successful or highly successful, which yields a contradiction. □

Similar to trust-region methods, the previous result can be used to prove a stronger statement for functions with a uniformly continuous gradient. The proof generalizes the one of [59, Theorem 3.5].

**Theorem 4.6.** *Let $\{B_k\}$ be a bounded sequence of symmetric matrices, assume that $\psi$ is bounded from below and that $\nabla f$ is uniformly continuous on a set $X$ satisfying $\{x^k\} \subset X$, where $\{x^k\}$ denotes a sequence generated by Algorithm 3.1. Then $\lim_{k\to\infty} \|r(x^k)\| = 0$ holds; in particular, every accumulation point of $\{x^k\}$ is a stationary point of $\psi$.*

*Proof.* Assume, by contradiction, that there exists $\delta > 0$ and $K \subset \mathbb{N}$ such that $\|r(x^k)\| \geq 2\delta$ for all $k \in K$. By Theorem 4.5, for each $k \in K$, there is an index $\ell(k) > k$ such that $\|r(x^l)\| \geq \delta$ for all $k \leq l < \ell(k)$ and $\|r(x^{\ell(k)})\| < \delta$.

If, for $k \in K$, an iteration $k \leq l < \ell(k)$ is successful or highly successful, we get

$$\psi(x^l) - \psi(x^{l+1}) \geq c_1 \operatorname{pred}_l \geq c_1 p_{\min}\|r(x^l)\| \cdot \|d^l\| \geq c_1 p_{\min}\delta\|x^{l+1} - x^l\|.$$

For unsuccessful iterations $l$, this estimate holds trivially. Thus,

$$p_{\min}c_1\delta\|x^{\ell(k)} - x^k\| \leq p_{\min}c_1\delta \sum_{l=k}^{\ell(k)-1} \|x^{l+1} - x^l\| \leq \sum_{l=k}^{\ell(k)-1} \psi(x^l) - \psi(x^{l+1}) = \psi(x^k) - \psi(x^{\ell(k)})$$

holds for all $k \in K$. By assumption, $\psi$ is bounded from below, and by construction, the sequence $\{\psi(x^k)\}$ is monotonically decreasing, hence convergent. This implies $\big\{\psi(x^k) -$

$\psi(x^{\ell(k)})\big\} \to_K 0$. Hence, we get $\big\{\|x^{\ell(k)} - x^k\|\big\} \to_K 0$. The uniform continuity of $\nabla f$ and of the proximity operator (Proposition 2.2 (a)) together with the fact that the composition of uniformly continuous functions is uniformly continuous, yields the uniform continuity of the residual function $r(\cdot)$. Thus, we get $\big\{\|r(x^{\ell(k)}) - r(x^k)\|\big\} \to_K 0$. On the other hand, by the choice of $\ell(k)$, we have

$$\big\|r(x^k) - r(x^{\ell(k)})\big\| \geq \big\|r(x^k)\big\| - \big\|r(x^{\ell(k)})\big\| \geq 2\delta - \delta \geq \delta,$$

which yields the desired contradiction. □

## 5 Convergence Using an Error Bound Condition

The aim of this section is to provide further convergence results for the regularized proximal quasi-Newton method in Algorithm 3.1. To this end, we start with some technical results and then assume that $\nabla f$ is Lipschitz continuous to show the boundedness of the sequence $\{\mu_k\}$. Together with an error bound condition, we then deduce the convergence of the entire sequence. We start with some technical results.

**Lemma 5.1.** *Assume that the sequence $\{H_k\}$ is uniformly bounded and positive definite, i.e. there exist constants $0 < m \leq M$ such that $mI \preceq H_k \preceq MI$ holds for all $k \geq 0$. Then the following estimates hold:*

(a) $\operatorname{pred}_k \geq \dfrac{1}{2}(m + 2\mu_k)\|d^k\|^2$,

(b) $\dfrac{\|r(x^k)\|}{\|d^k\|} \leq \Big(1 + \dfrac{1}{m + \mu_k}\Big)(M + \mu_k) \leq \dfrac{m+1}{m}(M + \mu_k)$,

(c) $\dfrac{\|d^k\|}{\|r(x^k)\|} \leq \dfrac{1 + M + \mu_k}{m + \mu_k} \leq \dfrac{1+M}{m}$.

*Proof.* (a) Using [36, Proposition 2.4], we get

$$\begin{aligned}\operatorname{pred}_k &= -\big(\nabla f(x^k)^T d^k + \varphi(x^k + d^k) - \varphi(x^k)\big) - \frac{1}{2}(d^k)^T H_k d^k \\ &\geq (d^k)^T (H_k + \mu_k I) d^k - \frac{1}{2}(d^k)^T H_k d^k \\ &\geq \frac{1}{2}(m + 2\mu_k)\|d^k\|^2.\end{aligned}$$

(b) and (c) follow directly from Lemma 2.3 using $\lambda_{\max}(H_k + \mu_k I) \leq M + \mu_k$ and $\lambda_{\min}(H_k + \mu_k I) \geq m + \mu_k$. □

The next result is essential to prove the boundedness of the sequence of regularizers $\{\mu_k\}$.

**Lemma 5.2.** *Assume that $\nabla f$ is Lipschitz continuous with Lipschitz constant $L > 0$ and $H_k \succeq mI$ for some $m > 0$. If, in some iterate $x^k$, we have $\mu_k \geq \overline{\mu} := \max\{L - m, 0\}$, there holds* $\operatorname{ared}_k > c_1 \operatorname{pred}_k$.

*Proof.* Let $\mu_k \geq \overline{\mu}$. Then $H_k + \mu_k I \succeq LI$, and the Lipschitz continuity of $\nabla f$ yields

$$f(x^k + d^k) - f(x^k) \leq \nabla f(x^k)^T d^k + \frac{1}{2}L\|d^k\|^2 \leq \nabla f(x^k)^T d^k + \frac{1}{2}(d^k)^T (H_k + \mu_k I) d^k,$$

which is equivalent to

$$\psi(x^k + d^k) - \psi(x^k) \le \nabla f(x^k)^T d^k + \varphi(x^k + d^k) - \varphi(x^k) + \frac{1}{2}(d^k)^T (H_k + \mu_k I) d^k.$$

Hence, using the definitions of $\mathrm{pred}_k$ and $\mathrm{ared}_k$, we get $-\mathrm{ared}_k \le -\mathrm{pred}_k + \mu_k/2\ \|d^k\|^2$. A combination with Lemma 5.1 (a) yields

$$\mathrm{ared}_k \ge \mathrm{pred}_k - \frac{\mu_k}{2}\|d^k\|^2 \ge \mathrm{pred}_k \cdot \frac{\mu_k + m}{2\mu_k + m} > \frac{1}{2}\,\mathrm{pred}_k \ge c_1\,\mathrm{pred}_k,$$

which had to be shown (note that we need $c_1 \le \frac{1}{2}$ at this point). □

For the boundedness of the sequence $\{\mu_k\}$, it remains to prove that (3.4) holds for sufficiently large $\mu_k > 0$, which is the aim of the next result.

**Proposition 5.3.** *Assume that $\nabla f$ is Lipschitz continuous with Lipschitz constant $L > 0$ and $MI \succeq H_k \succeq mI$ for some $M \ge m > 0$. Then, the sequence $\{\mu_k\}$ generated from Algorithm 3.1 is bounded.*

*Proof.* Assume that the sequence $\{\mu_k\}$ is unbounded. This means, there is a subsequence $K \subset \mathbb{N}_0$ such that $\{\mu_k\}_K \to \infty$. Since $\mu_k$ cannot increase in successful or highly successful steps, this implies that there are infinitely many unsuccessful steps. Without loss of generality we assume that all steps $k \in K$ are unsuccessful. In view of Lemma 5.2 this is only possible if for sufficiently large $k \in K$ we have $\mathrm{pred}_k < p_{\min}\|d^k\| \cdot \|r(x^k)\|$. Using Lemma 5.1 (a), this yields

$$\frac{m + 2\mu_k}{2}\|d^k\| < p_{\min}\|r(x^k)\| \qquad \Longleftrightarrow \qquad \frac{\|r(x^k)\|}{\mu_k\|d^k\|} > \frac{m + 2\mu_k}{2p_{\min}\mu_k}.$$

We combine this estimate with Lemma 5.1 (b) to get

$$\left(1 + \frac{1}{m + \mu_k}\right)\frac{M + \mu_k}{\mu_k} > \frac{m + 2\mu_k}{2p_{\min}\mu_k}$$

for $k \in K$. Taking the limit in $K$, the left hand side of this estimate converges to 1, whereas the right hand side converges to $1/p_{\min} > 1$, which yields a contradiction. Hence, the sequence $\{\mu_k\}$ is bounded. □

For the convergence of the complete sequence, we need an additional assumption. In many papers the main assumption to prove local convergence and state a convergence rate is strong convexity. Here, more generally, we assume that $\psi$ satisfies a local error bound condition, which is used by Tseng and Yun in [62].

**Assumption 5.1.** Assume that $\psi$ is bounded from below and $\mathcal{X}^* \ne \emptyset$, where $\mathcal{X}^*$ is the set of stationary points of $\psi$.

(a) For any $\zeta \ge \min_x \psi(x)$, there exist scalars $\tau > 0$ and $\varepsilon > 0$ such that

$$\mathrm{dist}(x, \mathcal{X}^*) \le \tau\|r(x)\| \quad \text{whenever} \quad \psi(x) \le \zeta,\ \|r(x)\| \le \varepsilon.$$

(b) There exists a scalar $\delta > 0$ such that

$$\|x - y\| \ge \delta \quad \text{whenever} \quad x \in \mathcal{X}^*, y \in \mathcal{X}^*, \psi(x) \ne \psi(y).$$

Similar assumptions to (a) have been investigated by Luo and Tseng in [42, 43]. Note that if a function satisfies the above error bound condition, then it also satisfies the Kurdyka-Łojasiewicz property [38]. Error bounds of this type have been studied by many authors, see e.g. [67, 68].

Some examples of problem classes of the form (1.1) that satisfy Assumption 5.1 (a) are, cf. [62, 67] and the references therein:

- The function $f$ is strongly convex, $\nabla f$ is Lipschitz continuous and $\varphi$ is an arbitrary convex function.
- $f(x) = h(Ax) + c^T x$, where $h : \mathbb{R}^m \to \mathbb{R}$ is a continuously differentiable and strongly convex function such that $\nabla h$ is Lipschitz continuous on every compact set, $A \in \mathbb{R}^{m\times n}, c \in \mathbb{R}^n$, and $\varphi$ has a polyhedral epigraph.
- $f(x) = h(Ax)$, where $A \in \mathbb{R}^{m\times n}$ and $h$ is given as above, and $\varphi(x) = \sum_{i=1}^{s} \|x_{G_i}\|_2$, where the sets $G_i \subset \{1, \ldots, n\}$ form a partition of $\{1, \ldots, n\}$ .

Many more functions of type (1.1) fulfill Assumption 5.1 (a) even if they are not covered by the above problem classes. For more information and properties of error bound conditions, we refer to [62, 67, 68].

Assumption 5.1 (b) guarantees that the sets of stationary points of $\psi$ with different function values are properly separated. This assumption holds, in particular, if $\psi$ is convex.

It is important to note that we do not assume the convergence of the sequence $\{x^k\}$. Instead, this is a consequence of the above assumptions, as the following result shows.

**Theorem 5.4.** *Let $\{x^k\}$ be a sequence generated by Algorithm 3.1 such that $\nabla f$ is Lipschitz continuous, $MI \succeq H_k \succeq mI$ for some $M \geq m > 0$, and let Assumption 5.1 hold. Then the sequence $\{x^k\}$ converges to some $\overline{x} \in \mathbb{R}^n$ and $\sum_{k=0}^{\infty} \|x^{k+1} - x^k\| < \infty$.*

This result is a simplified version of Theorem 2 in [62] and, therefore, we skip the proof here. However, we briefly discuss the essential adaptations: First, the estimate of Lemma 5.1 (c) in combination with Theorem 4.6 yields $d^k \to 0$. Moreover, the crucial preliminary of [62, Theorem 2] is the boundedness of the analogous sequence to $\{B_k + \mu_k I\}$, which in our analysis is the result of the assumption on $\{B_k\}$ and Proposition 5.3. The further details of the proof are left to the reader.

We note that it is also possible to develop a local convergence theory for Algorithm 3.1 with small adjustments and under appropriate assumptions. In this paper, we focus on limited memory quasi-Newton approximations and therefore focus on the efficient solution of the related subproblems, which is the topic of the next section.

## 6 Application to Limited Memory Proximal Quasi-Newton Methods

This section describes the central part for an efficient implementation of Algorithm 3.1 using limited memory matrices for $B_k$. Since the idea itself is central for our work, we first present the basic steps in a slightly simplified framework in Section 6.1, and then come to the details for the actual realization in Section 6.2.

### 6.1 Main Idea Based on Compact Representations

The most costly part of Algorithm 3.1 is the computation of $d^k$ in (S.2), which requires the solution of the minimization problem

$$\min_d f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T (B_k + \mu_k I) d + \varphi(x^k + d). \tag{6.1}$$

In the following, we assume that the matrix $B_k + \mu_k I$ is positive definite to ensure that the problem is solvable. If this is not the case, the problem might be unsolvable (depending on the properties of $\varphi$). Nevertheless, the following explanation mainly considers quasi-Newton matrices $B_k$ which fulfil this requirement under mild assumptions. If these are not met, the update is skipped.

So, if $B_k + \mu_k I$ is positive definite, we use the proximity operator to reformulate the problem to

$$d^k = r_{B_k+\mu_k I}(x^k) = \operatorname{prox}_{\varphi}^{B_k+\mu_k I}\left(x^k - (B_k + \mu_k I)^{-1} \nabla f(x^k)\right) - x^k. \tag{6.2}$$

Hence, the main effort is the computation of the proximity operator with respect to the norm induced by $B_k + \mu_k I$, where we are especially interested in the case that $B_k$ is obtained using a limited memory quasi-Newton update. Since this is usually not possible analytically, appropriate algorithms must be used for this computation. A general approach is to apply first-order proximal methods like FISTA [4] for the solution of the subproblem, cf. [29, 36]. Further methods as [6,22,30] exploit the structure of limited memory quasi-Newton matrices, but these methods are only considered for memoryless updates.

The crucial point of our method to compute the solution of (6.1) consists in a suitable combination of a recent result by Becker et al. [6] with the compact representation of limited memory quasi-Newton matrices introduced by Byrd et al. [12]. We first describe the idea of our approach, and then provide the corresponding details for the actual realization (implementation) of the resulting method.

The class of quasi-Newton methods generates a sequence $\{x^k\}$ using the recursion $x^{k+1} := x^k - H_k^{-1} \nabla f(x^k)$ for some suitable approximation $H_k$ of the (not necessarily existing) Hessian $\nabla^2 f(x^k)$ (in our setting, we have $H_k = B_k + \mu_k I$). The matrices $H_k$ are usually updated using rank-one or rank-two modifications; two well-known examples are the SR1 update (symmetric rank-one)

$$H_{k+1} := H_{k+1}^{SR1} := H_k + \frac{(y^k - H_k s^k)(y^k - H_k s^k)^T}{(y^k - H_k s^k)^T s^k}$$

and the BFGS update (Broyden-Fletcher-Goldfarb-Shanno)

$$H_{k+1} := H_{k+1}^{BFGS} := H_k + \frac{y^k (y^k)^T}{(s^k)^T y^k} - \frac{H_k s^k (s^k)^T H_k}{(s^k)^T H_k s^k},$$

where

$$s^k := x^{k+1} - x^k, \quad y^k := \nabla f(x^{k+1}) - \nabla f(x^k) \quad \forall k \in \mathbb{N}.$$

These quasi-Newton methods are not applicable to large-scale problems since the matrices $H_k$ are dense. This problem can be avoided based on the following observation: The matrix $H_{k+1}$ can, in principle, be re-computed using the data $H_0$ together with the vectors $s^j$ and $y^j$ for all $j = 1, 2, \ldots, k$. Now, if we skip the first of these vectors and use only the final $m$ ones (for some small memory $m \in \mathbb{N}$), we obtain a limited memory quasi-Newton method, cf. [52], which, due to a much smaller storage requirement, can be applied to large-scale problems.

These limited memory versions of standard quasi-Newton updates, however, may not start with the same initial matrix $H_0$, instead they often use an initialization $H_{k,0}$ depending on the current iterate $k$.

Now, consider the proximal subproblem

$$\min_d f(x^k) + \nabla f(x^k)^T d + \tfrac{1}{2} d^T H_k d + \varphi(x^k + d) \tag{6.3}$$

for some suitable matrix $H_k$. Using $H_k := \lambda_k I$ ($\lambda_k > 0$), this subproblem is often easy to solve (sometimes even analytically), whereas we obtain a much better approximation of the given composite optimization problem if $H_k$ is chosen as a better approximation of the Hessian $\nabla^2 f(x^k)$, but then the subproblem itself is more difficult to solve. However, if

$$H_k = H_{k,0} + U_1 U_1^T - U_2 U_2^T \tag{6.4}$$

with suitable matrices $U_i \in \mathbb{R}^{n \times r_i}$ (usually depending on $k$, but to simplify the notation, we skip this index here) for some small $r_i \in \mathbb{N}$ ($i = 1, 2$) and a simple matrix $H_{k,0}$ (typically a multiple of the identity matrix such that the corresponding proximal subproblem is easy to solve), so that $H_k$ is obtained from $H_{k,0}$ by a small rank-modification, then it is shown in Becker et al. [6] that the solution of the difficult subproblem (6.3) can be computed from the solution of the (easy) proximal subproblem corresponding to the matrix $H_{k,0}$ using only some matrix-vector multiplications and solving a (strongly monotone, hence uniquely solvable) nonlinear system of equations of (small) dimension $r_1 + r_2$.

Recalling the typical updates of quasi-Newton matrices, we immediately see that a single update of, e.g., the SR1- and the BFGS-method is precisely of the form required in (6.4) with suitable matrices $U_1, U_2$ of rank (at most) one. However, since the additive terms in these quasi-Newton updates depend on $H_k$ itself, these formulas cannot be used (directly) to apply the result from [6], which is based on the representation (6.4), to limited memories with $m \geq 2$. In fact, numerical results presented in [6] are based on taking a limited memory of $m = 1$ only. Their point is that for $m = 1$ in the SR1-update, the occuring nonlinear system is of dimension 1 and can, hence be solved by bisection, and, if $\varphi$ is piecewise linear, even exact in log-linear time.

For many medium-sized problems, however, there are advantages to use a memory larger than 1. This is the point where we can use the so-called compact representations of limited memory quasi-Newton matrices.

The Hessian approximations generated by most limited memory quasi-Newton methods can be written using a compact representation of the form

$$H_k = H_{k,0} + A_k Q_k^{-1} A_k^T \tag{6.5}$$

for some (usually diagonal) symmetric positive definite matrix $H_{k,0} \in \mathbb{R}^{n \times n}$, $A_k \in \mathbb{R}^{n \times s}$, and a symmetric and nonsingular matrix $Q_k \in \mathbb{R}^{s \times s}$, where, again, $s \ll n$ is typically a very small number. Such a compact representation can be used in order to rewrite $H_k$ in a form required in (6.4). To this end, we compute a spectral decomposition $Q_k^{-1} = V_k \Lambda_k V_k^T$ of $Q_k^{-1}$, i.e., $V_k \in \mathbb{R}^{s \times s}$ is orthogonal and $\Lambda_k \in \mathbb{R}^{s \times s}$ is a diagonal matrix with diagonal entries $\lambda_i^k$ (recall that $s$ is small, hence the computation of this spectral decomposition is not at all time-consuming). We then split the diagonal matrix $\Lambda_k$ into

$$\Lambda_k = \Lambda_k^+ - \Lambda_k^-,$$

where $\Lambda_k^+$ and $\Lambda_k^-$ are diagonal matrices consisting of the elements $\max\{0, \lambda_i^k\}$ and $\max\{0, -\lambda_i^k\}$, respectively. Note that this implies that these two diagonal matrices are

positive semidefinite and, therefore, possess a matrix square root. Substituting this into (6.5) yields the representation (6.4) with the matrices (their dependence on $k$ is neglected here)

$$U_1 := A_k V_k (\Lambda_k^+)^{1/2} \quad \text{and} \quad U_2 := A_k V_k (\Lambda_k^-)^{1/2}.$$

Note that the two matrices $U_1, U_2$ actually simplify to some extent since some of their columns are multiplied with zero entries of the corresponding diagonal matrices. This completes the general description which allows an efficient implementation of our regularized proximal limited memory quasi-Newton method.

## 6.2 Realization of Proximal Subproblem Solutions

We now present the details of our realization of Algorithm 3.1 where, we recall, we have $H_k = B_k + \mu_k I$ in the notation of the previous subsection, and where we use a limited memory update of $B_k$ (not of $H_k$ itself), whereas the regularization term essentially only influences the initial matrix $H_{k,0}$ (or $B_{k,0}$ in our subsequent notation) since, in any case, this is typically just a multiple of the identity matrix. Hence, assume we have a compact representation of the form

$$B_k = B_{k,0} + A_k Q_k^{-1} A_k^T,$$

where $B_{k,0} \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, usually chosen as a multiple of the identity, $Q_k \in \mathbb{R}^{s \times s}$ is a symmetric and nonsingular matrix with $s \ll n$, and $A_k \in \mathbb{R}^{n \times s}$, cf. [12]. The following example states explicitly the compact representations of the SR1- and the BFGS-updates, since these two will be exploited in our numerical experiments.

**Example 6.1.** As before, let $s^j = x^{j+1} - x^j$ and $y^j = \nabla f(x^{j+1}) - \nabla f(x^j)$ for all $j$. Then, in iteration $k$, we define the matrices

$$S_k := [s^{k-m} \dots s^{k-1}] \in \mathbb{R}^{n \times m} \quad \text{and} \quad Y_k := [y^{k-m} \dots y^{k-1}] \in \mathbb{R}^{n \times m}.$$

Furthermore, let $D_k = D(S_k^T Y_k)$ and $L_k = L(S_k^T Y_k)$ denote the diagonal part and the strict lower triangle of the matrix $S_k^T Y_k$. Then, the corresponding limited memory BFGS-update is given by the compact representation

$$B_k := B_k^{BFGS} = B_{k,0} - \begin{bmatrix} B_{k,0} S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^T B_{k,0} S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T B_{k,0} \\ Y_k^T \end{bmatrix},$$

hence,

$$A_k = \begin{bmatrix} B_{k,0} S_k & Y_k \end{bmatrix} \in \mathbb{R}^{n \times 2m} \quad \text{and} \quad Q_k = \begin{bmatrix} -S_k^T B_{k,0} S_k & -L_k \\ -L_k^T & D_k \end{bmatrix} \in \mathbb{R}^{2m \times 2m}.$$

Similarly, the limited memory SR1-update can be written as

$$B_k := B_k^{SR1} = B_{k,0} + (Y_k - B_{k,0} S_k)(D_k + L_k + L_k^T - S_k^T B_{k,0} S_k)^{-1} (Y_k - B_{k,0} S_k)^T,$$

which yields

$$A_k = Y_k - B_{k,0} S_k \in \mathbb{R}^{n \times m} \quad \text{and} \quad Q_k = D_k + L_k + L_k^T - S_k^T B_{k,0} S_k \in \mathbb{R}^{m \times m},$$

see [12, Theorems 2.3 and 5.1]. ◇

To simplify the following discussion, we consider a fixed iteration $k$ and therefore omit this index in the subsequent notation.

Similar to Section 6.1, with the matrix $Q = Q_k$ available from the compact representation, we then compute a spectral decomposition $Q^{-1} = V\Lambda V^T$ of $Q^{-1}$ with $V \in \mathbb{R}^{s\times s}$ being orthogonal and $\Lambda \in \mathbb{R}^{s\times s}$ being a diagonal matrix. Let $\mathcal{I}_1, \mathcal{I}_2 \subset \{1, 2, \ldots, s\}$ be the sets of indices corresponding to the positive and negative entries of the diagonal of $\Lambda$, respectively.

Define $\Lambda_1$ as the submatrix of $\Lambda$ with the rows and columns in $\mathcal{I}_1$ and $\Lambda_2$ as the submatrix of $-\Lambda$ with the rows and columns in $\mathcal{I}_2$, and let $(AV)_1, (AV)_2$ be the submatrices of $A \cdot V$ with the column indices in $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. Then we can write

$$B = B_0 + U_1 U_1^T - U_2 U_2^T$$

with

$$U_1 := (AV)_1 \Lambda_1^{1/2} \quad \text{and} \quad U_2 := (AV)_2 \Lambda_2^{1/2}. \tag{6.6}$$

Note that, by defining $\widehat{B_0} = B_0 + \mu I$, we obtain a similar formula for the matrix $\widehat{B} = B + \mu I$. At this point, we can use the following result from [6, Corollary 3.6] for the solution of (6.2).

**Theorem 6.2.** *Let $\widehat{B} = \widehat{B}_0 + U_1 U_1^T - U_2 U_2^T \in \mathbb{S}^n_{++}$ with $\widehat{B}_0 \in \mathbb{S}^n_{++}$ and $U_i \in \mathbb{R}^{n\times r_i}$ with rank $r_i$ $(i = 1, 2)$. Set $\widehat{B}_1 = \widehat{B}_0 + U_1 U_1^T$. Then, the following holds:*

$$\operatorname{prox}^{\widehat{B}}_{\varphi}(y) = \operatorname{prox}^{\widehat{B}_0}_{\varphi}(y + \widehat{B}_1^{-1} U_2 \alpha_2^* - \widehat{B}_0^{-1} U_1 \alpha_1^*), \tag{6.7}$$

*where $\alpha_i^* \in \mathbb{R}^{r_i}$, $i = 1, 2$, are the unique zeros of the coupled system $\mathcal{L}(\alpha) = \mathcal{L}(\alpha_1, \alpha_2) = 0$, where $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2)$ is defined by*

$$\begin{aligned}
\mathcal{L}_1(\alpha_1, \alpha_2) &= U_1^T(y + \widehat{B}_1^{-1} U_2 \alpha_2 - \operatorname{prox}^{\widehat{B}_0}_{\varphi}(y + \widehat{B}_1^{-1} U_2 \alpha_2 - \widehat{B}_0^{-1} U_1 \alpha_1)) + \alpha_1, \\
\mathcal{L}_2(\alpha_2, \alpha_2) &= U_2^T(y - \operatorname{prox}^{\widehat{B}_0}_{\varphi}(y + \widehat{B}_1^{-1} U_2 \alpha_2 - \widehat{B}_0^{-1} U_1 \alpha_1)) + \alpha_2.
\end{aligned} \tag{6.8}$$

In the following, we restrict the analysis to the case $B_0 = \gamma I$ for some $\gamma > 0$. Hence, in Theorem 6.2 we have $\widehat{B}_0 = \hat{\gamma} I$ with $\hat{\gamma} = \gamma + \mu$, which can be easily inverted and the proximity operator $\operatorname{prox}^{\widehat{B}_0}_{\varphi}$ can often be computed analytically. For the computation of $\widehat{B}_1^{-1}$ and $\widehat{B}^{-1}$, we use the Sherman-Morrison-Woodbury formula to obtain

$$\begin{aligned}
\widehat{B}_1^{-1} &= \hat{\gamma}^{-1} I - \hat{\gamma}^{-2} U_1 (I + \hat{\gamma}^{-1} U_1^T U_1)^{-1} U_1 \qquad \text{and} \\
\widehat{B}^{-1} &= \widehat{B}_1^{-1} + \widehat{B}_1^{-1} U_2 (I - U_2^T \widehat{B}_1^{-1} U_2)^{-1} U_2^T \widehat{B}_1^{-1}.
\end{aligned}$$

Since the proximity operator is Lipschitz continuous, nonsmooth (semismooth) Newton methods are suitable candidates for the numerical computation of the unique zero $\alpha^* = (\alpha_1^*, \alpha_2^*)$ of the nonlinear system of equations $\mathcal{L}(\alpha) = 0$ in Theorem 6.2. An iteration of the semismooth Newton method is given by

$$\alpha^{j+1} = \alpha^j - G_j^{-1} \mathcal{L}(\alpha^j), \tag{6.9}$$

where $G_j = G(\alpha^j)$ is a Newton derivative of $\mathcal{L}$ in $\alpha^j$, cf. [55]. For some details on Newton differentiable functions, we refer to [25]. Provided that the Newton derivative of the proximity operator can be computed, a short calculation and the chain rule for generalized derivatives [25, Theorem 3.5] show the following result.

**Proposition 6.3.** *Let* $\operatorname{prox}_{\varphi}^{\widehat{B}_0}$ *be Newton-differentiable with generalized derivative* $P$*. Then* $\mathcal{L}$ *is also Newton-differentiable, and the generalized derivative is given by*

$$G(\alpha) = \begin{bmatrix} U_1 & U_2 \end{bmatrix}^T P(z) \begin{bmatrix} \widehat{B}_0^{-1} U_1 & -\widehat{B}_1^{-1} U_2 \end{bmatrix} + \begin{bmatrix} I & U_1^T \widehat{B}_1^{-1} U_2 \\ 0 & I \end{bmatrix}, \tag{6.10}$$

*where* $z = y + \widehat{B}_1^{-1} U_2 \alpha_2 - \widehat{B}_0^{-1} U_1 \alpha_1$.

In many applications the generalized derivative of the proximity operator can be computed analytically.

**Example 6.4.** (a) Let $\varphi(x) := \lambda \|x\|_1$ and $\widehat{B}_0 = \hat{\gamma} I$ for some $\lambda, \hat{\gamma} > 0$. Then the proximity operator is given (component-wise) by

$$\left(\operatorname{prox}_{\varphi}^{\hat{\gamma} I}\right)_i (x) = \begin{cases} x_i - \lambda\hat{\gamma}, & \text{if } x_i \geq \lambda\hat{\gamma}, \\ 0, & \text{if } |x_i| < \lambda\hat{\gamma}, \\ x_i + \lambda\hat{\gamma}, & \text{if } x_i \leq -\lambda\hat{\gamma}, \end{cases}$$

cf. [45, Example 3.2.8]. Hence, the diagonal matrix $P(x)$ with diagonal entries

$$P_{ii}(x) = \begin{cases} 1, & \text{if } |x_i| \geq \lambda\hat{\gamma}, \\ 0, & \text{otherwise} \end{cases}$$

is an element of the generalized Jacobian in the sense of Clarke, cf. [16], and, therefore, a Newton derivative.

(b) Let $\varphi(x) := \lambda \|x\|_2$. Then, an elementary calculation shows

$$\operatorname{prox}_{\varphi}^{\hat{\gamma} I}(x) = x \cdot \max\left\{1 - \frac{\lambda\hat{\gamma}}{\|x\|_2}, 0\right\}$$

cf. [45, Example 3.2.8]. A short computation therefore shows that the following is a Newton derivative of this proximity operator:

$$P(x) = \begin{cases} \left(1 - \frac{\lambda\hat{\gamma}}{\|x\|_2}\right) I + \frac{\lambda\hat{\gamma}}{\|x\|_2^3} x x^T, & \text{if } \|x\|_2 \geq \lambda\hat{\gamma}, \\ 0, & \text{otherwise.} \end{cases}$$

The two examples given here will be used in our numerical section. ◇

We summarize the previous discussion and present our method for the computation of (6.2) in the following algorithm.

**Algorithm 6.5** (Solution of the subproblem (6.2))**.**

(S.0) Given an iterate $x^k$, a compact representation $B_k = \gamma_k I + A_k Q_k^{-1} A_k^T$ of the corresponding Hessian approximation, and $\mu_k > 0$.

(S.1) Compute the spectral decomposition $Q_k^{-1} = V_k \Lambda_k V_k^T$, define

$$\mathcal{I}_1 := \left\{i \in \{1, \ldots, s\} \mid \Lambda_k(i,i) > 0\right\}, \qquad \mathcal{I}_2 := \left\{i \in \{1, \ldots, s\} \mid \Lambda_k(i,i) < 0\right\},$$

and determine $U_1, U_2$ according to (6.6).

(S.2) Choose $\alpha^0 \in \mathbb{R}^{r_1 + r_2}$ and compute the zero $\alpha^*$ of $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2)$ defined in (6.8), using a semismooth Newton method with the updates given in (6.9) and the generalized Jacobian given in (6.10), until a suitable termination criterion holds.

(S.3) Compute $d^k = \operatorname{prox}_{\varphi}^{B_k + \mu_k I}\left(x^k - (B_k + \mu_k I)^{-1} \nabla f(x^k)\right) - x^k$ using (6.7).

Of course, the most expensive part of Algorithm 6.5 is the solution of the semismooth Newton equation in (S.2). While Becker et al. [6] suggest a solution using an inexact semismooth

Newton method in the general case, our experiments show that using the above described method performs just a few (in most cases 1-2) iterations to end up with an approximation of $\alpha^*$ satisfying $\|\mathcal{L}(\alpha^*)\| < 10^{-10}$ independently of the size of the memory. This underlines the high efficiency of Algorithm 6.5, in particular using memories larger than one.

## 7 Numerical Results

In this section, we report numerical results for solving problem (1.1) using the Regularized Proximal Quasi-Newton Method (RPQN) from Algorithm 3.1 with limited memory quasi-Newton matrices. After comparing different limited memory methods for the computation of the occuring proximity operators, we compare this method with several methods applicable to solve problem (1.1).

The numerical results have been obtained in MATLAB R2020b using a machine running Open SuSE Leap 15.2 with an Intel Core i5 processor 3.2 GHz and 16 GB RAM.

### 7.1 Least Squares Problems with Group Sparse Regularizer

In our first example, we consider the least squares problem for $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ with an $\ell_1$-$\ell_2$-sparsity regularizer, which is also called a group sparse regularizer in the literature. The problem is given by

$$\min_x \frac{1}{2}\big\|Ax - b\|_2^2 + \lambda\|x\|_{2,1},$$

where

$$\|x\|_{2,1} := \sum_{j=1}^{p} \|x_{\mathcal{I}_j}\|_2.$$

Here, the index sets $\mathcal{I}_j$ $(j = 1, \dots, p)$ form a partition of $\{1, \dots, n\}$. Since the groups $\mathcal{I}_j$ are pairwise disjoint, the proximity operator $\operatorname{prox}_{\lambda\|\cdot\|_{2,1}}$ and a Newton derivative thereof can be computed block-wise using the formulas in Example 6.4. The use of the $\ell_1$-$\ell_2$-regularizer makes sense in many applications, where sparsity should be achieved with respect to some groups of variables. We refer to [44] for more information about group (sparse) regularizers.

Note that the gradient $\nabla f(x) = A^T(Ax - b)$ of the function $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ is obviously Lipschitz continuous. Hence, the assumptions of Theorem 4.6 are satisfied. Furthermore, by discussion in Section 5, this problem setting also satisfies Assumption 5.1 which, due to the convexity of the problem setting, implies the convergence of the complete sequence to a global minimizer.

#### 7.1.1 Problem Setting and Implementation

We follow the generic example in [6] and choose the entries in $A$ and $b$ from a uniform distribution in $[0, 1]$ with $n = 25k$ and $m = 16k$ for various $k \in \mathbb{N}$. The parameter $\lambda$ is set to 1. Furthermore, the index sets $\mathcal{I}_j$ are chosen randomly with 4 to 12 elements. The initial guess for the iterate is $x^0 = 0$. In Algorithm 3.1, we choose the parameters $\mu_0 = 1$, $p_{\min} = c_1 = 10^{-4}$, $c_2 = 0.9$, $\sigma_1 = 0.5$ and $\sigma_2 = 4$. These values are quite typical: $p_{\min} > 0$ should be taken small in order to guarantee that many iterations are successful or highly successful. For the same reason, $c_1$ is a small positive constant, whereas a practical choice for $c_2$ in the related context of trust-region methods is usually some number from the interval $[0.7, 0.9]$. The values of $\sigma_1$ and $\sigma_2$ are also motivated from the corresponding updates used for trust-region methods.

Furthermore, our tests showed that the semismooth Newton method for the computation of the proximity operators in Algorithm 6.5 converges very fast (mostly within 1 or 2 steps), so we stop if $\|\mathcal{L}(\alpha)\| < 10^{-10}$ and use a maximal iteration number of 10. Since the limited memory BFGS-updates are only well-defined if $(s^k)^T y^k > 0$, it is common to skip the update of the limited memory matrices if $(s^k)^T y^k < \varepsilon \|s^k\|^2$. For the SR1-update ill-conditioned steps are skipped easily in a similar way as described in [12]: Instead of computing the spectral decomposition of $Q_k^{-1}$ in Algorithm 6.5, we compute the spectral decomposition $V_k \Lambda_k V_k^T$ of $Q_k$ and define the index sets $\mathcal{I}_1$ and $\mathcal{I}_2$ to contain the indices such that $\Lambda_k(i,i) > \varepsilon$ and $\Lambda_k(i,i) < -\varepsilon$, respectively. With this strategy, rows and columns with ill-conditioned steps ($|\Lambda_k(i,i)| \leq \varepsilon$) are skipped. We choose $\varepsilon = 10^{-8}$ in our experiments and note that updates are almost never skipped. The initial estimate $\gamma_k$ for the computation of the limited memory quasi Newton matrices is set to

$$\gamma_k = \frac{(y^k)^T y^k}{(s^k)^T y^k},$$

following the approach of Liu and Nocedal [40]. There are several ways to update the matrix $B_k$ if a step was unsuccessful. In this case one could start again with memory 0. However, our experiments show better results if the update of $B_k$ is simply skipped.

An obvious termination criterion would be the size of $\|r(x^k)\|$. However, this is less suitable for comparing different methods, as this quantity is not computed automatically by other methods, which then leads to an additional computational effort. Thus, to compare different methods, we initially run the algorithm once with a very high accuracy to determine a good approximation to the optimal function value $\psi^*$, and then terminate the methods if the current iterate $x^k$ satisfies

$$\frac{\psi(x^k) - \psi(x^*)}{\max(1, |\psi(x^*)|)} \leq 10^{-6}, \tag{7.1}$$

where the term on the left hand side is referred to as *objective value error*. Besides analysing the regularized proximal quasi-Newton method (RPQN) itself, we compare it to the following methods:

- QGPN (Globalized Proximal Quasi-Newton Method [29])

  This method represents a class of several proximal quasi-Newton methods, which use an Armijo-type line search strategy to guarantee convergence. In contrast to other methods, e.g. [6,36], a further globalization using a proximal gradient method is applied, which has shown to improve the performance. Parameters are chosen as in [29].

In addition to this second order proximal method, we use two well known proximal first order methods to compare the results to. Although there are plenty of accelerated proximal first order methods, to the author's knowledge there is no clear favourite regarding the performance. Hence, we chose the following well-known ones.

- FISTA (Fast Iterative Shrinkage Thresholding Algorithm [4])

  FISTA is one of the most common accelerated first order methods for solving convex problems with composite functions. In every step a subproblem of the form (6.2) is solved, where $B_k + \mu_k I$ is replaced by $L_k I$ and $L_k$ is an approximation to the Lipschitz constant of $\nabla f$. We start with the initial guess $L_0 = 1$ and increase with $\eta = 2$, if the

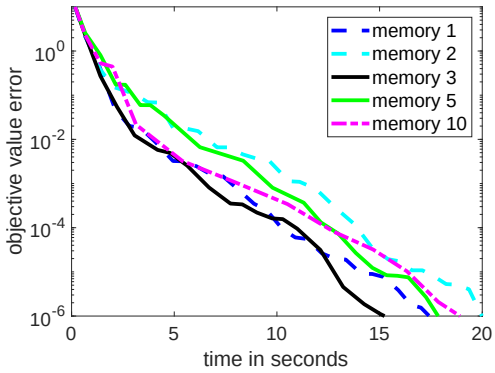



Figure 1: Convergence plot for RPQN with limited memory BFGS approach and different memories for the setting in Section 7.1 for solving the least squares problem with group sparse regularizer. The run time is the average of 10 runs.

step is not successful.
Although there are several adaptations of FISTA in the nonconvex setting, e.g. [53], we restrict the analysis to the convex version.

- SpaRSA (Sparse Reconstruction by Separable Approximation [66])

  SpaRSA is another first order method for the considered problem class. The main difference to FISTA is the update of the factor $L_k$, which is done by a Barzilai-Borwein approach. Hence, the method is related to RPQN with a memory of 0. Furthermore, the theory of SpaRSA also includes nonconvex functions.

All that techniques are proximal-type methods, since these are highly efficient for solving optimization problems with composite functions. In the above setting, we also tested a method based on the forward backward envelope [61]. Furthermore, the setting in the subsequent section allows using an interior point method, cf. [32]. However, these methods did not yield benefits in comparison to the above mentioned methods. Instead, we also provide comparisons with the following non-proximal method.

- SNF (Semismooth Newton Method with Multidim. Filter Globalization [45, 46])

  This method by Milzarek and Ulbrich is based on the semismooth Newton method to find a zero of $r(x)$, combined with a globalization using a filter strategy. There is a convex and nonconvex version of the filter conditions to decide whether the computed update is applied or a proximal gradient step is performed instead.

### 7.1.2 Discussion of the Results

We start comparing the size of the memory using the dimension $k = 100$, i.e. $n = 2500$ and $m = 1600$, which should be chosen for the limited memory quasi-Newton method. Figure 1 shows the relation between the elapsed run time and the current error as defined in (7.1), when RPQN is applied to the test problem with limited memory BFGS-updates. To avoid side effects and first-time computation costs, the time is averaged over 10 runs. The choice of the memory size should be big enough to achieve good performance, but preferably small to save computation costs. Figure 1 indicates that the impact of the memory size to the run time is relatively small, but the memory 3 showed the best performance. This is also validated by the data given in Table 1. In a similar test with limited memory SR1-updates, the best results were achieved with a memory of 5. RPQN with limited memory BFGS- and

SR1-updates and the determined optimal memory sizes are denoted by RPQN (L-BFGS) and RPQN (L-SR1), respectively.

For a comparison to other state-of-the-art methods, we take $k \in \{1, 3, 10, 30, 100, 300\}$ and run all algorithms on 10 random examples as described above. The average computation time in relation to the problem dimension is visualized in Figure 2. For the comparison we used RPQN and QGPN with limited memory BFGS-updates and a memory of 10. Note that QGPN did not converge within $10^4$ (outer) iterations for $n = 7500$. One sees that the performance of the first-order methods is better for small problem sizes. This follows from the high computation costs for solving the subproblems, which does not yield a profit for small dimensions. On the other hand, starting with $n = 750$, RPQN clearly outperforms the other methods, not only first-order, but also the tested second-order methods. This shows that the regularization in Algorithm 3.1 is superior although some iterations are unsuccessful and the computed solutions of the corresponding subproblems are discarded.

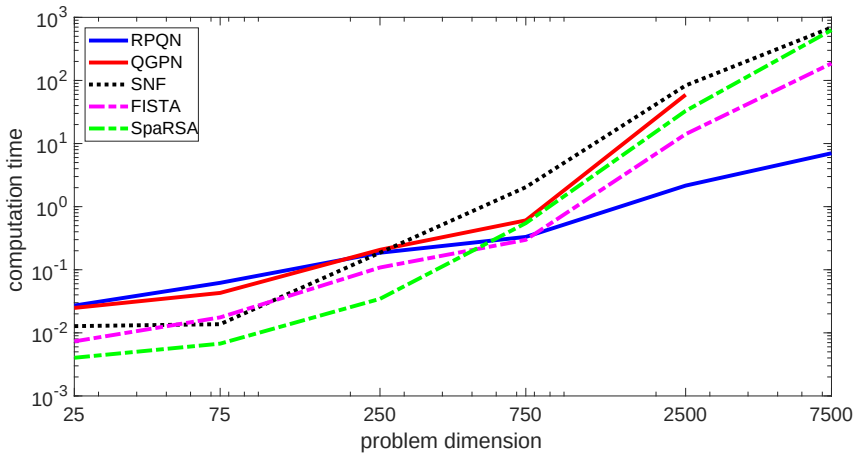



Figure 2: Comparison of the performance of several methods depending on the problem dimension as described in Section 7.1.

## 7.2 $\ell_1$-regularized Least Squares Problem (LASSO)

We demonstrate the performance of our method for the unconstrained LASSO (least absolute shrinkage and selection operator) problem

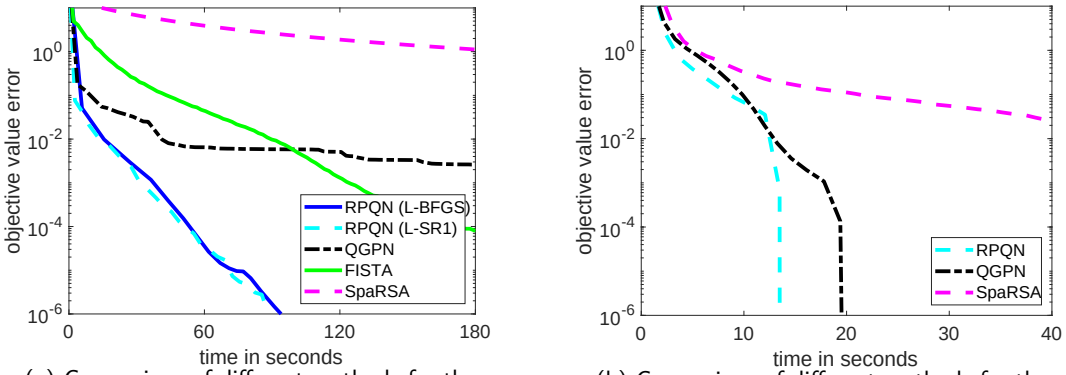
$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1,$$

| method (memory) | iter | highly s. iter | succ. iter | unsucc. iter | sub-iter | function eval | proximity eval | matrix-vector products |
|---|---|---|---|---|---|---|---|---|
| L-BFGS (1) | 46 | 18 | 14 | 14 | 199 | 47 | 442 | 94 |
| L-BFGS (2) | 36 | 18 | 5 | 13 | 149 | 36 | 333 | 73 |
| L-BFGS (3) | 49 | 27 | 6 | 16 | 208 | 50 | 461 | 100 |
| L-BFGS (5) | 55 | 32 | 3 | 20 | 265 | 53 | 577 | 106 |
| L-BFGS (10) | 34 | 20 | 2 | 12 | 121 | 33 | 276 | 66 |

Table 1: Values of the test example in Section 7.1 for the RPQN method with limited memory BFGS update and various memories.

with $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ and $\lambda > 0$. This formulation is used for many problems to handle sparsity when finding a solution of $Ax \approx b$, see e.g. [4, 19]. Again, we use a test setting from [6] with $n = 3000$ and $m = 1500$, which is typical for compressed sensing. The entries of $A$ and $b$ are independently and identically distributed according to the standard normal distribution, the penalty parameter is chosen as $\lambda = 0.1$. We use the methods described in Section 7.1 and almost all parameters are used as before, except that the memory for RPQN (L-BFGS) is set to $m = 10$, and QGPN is applied with a limited memory BFGS-update and a memory of 5, as these proved to be the best choices in our tests.

The results are illustrated in Figure 3 (a). Again one sees that there is almost no difference between the optimal versions (concerning the size of the memory) of the limited memory BFGS- and SR1-updates of RPQN. Furthermore, these methods perform significantly better than the other tested methods. While QGPN can keep up until an objective value error of approximately $10^{-1}$, its performance gets very slow afterwards. The first order methods FISTA and SpaRSA have by far longer running times to achieve appropriate errors.




(a) Comparison of different methods for the example in Section 7.2.

(b) Comparison of different methods for the example in Section 7.3.

Figure 3: Convergence plots for the $\ell_1$-regularized least squares problem (a) and the nonconvex image restoration (b). The run time is the average of 10 runs, the term "objective value error" refers again to the stopping criterion defined in (7.1).

### 7.3 Nonconvex Image Restoration

In this section, we consider a nonconvex image restoration problem. Given a noisy blurred image $b \in \mathbb{R}^n$ and a blur operator $A \in \mathbb{R}^{n\times n}$, the aim is to restore the original image $x \in \mathbb{R}^n$ such that $Ax \approx b$. If there are Gaussian errors on the image $b$, this problem can be solved efficiently using a quadratic loss similar to the previous sections. If the errors are distributed by Student's $t$-distribution, cf. [1], this approach usually does not perform well. For that purpose, the quadratic loss can be replaced by

$$f(x) := \sum_{i=1}^{n} \log\left((Ax - b)_i^2 + 1\right),$$

cf. [60]. To guarantee antialiasing, we add the nonsmooth term $\varphi(x) := \lambda\|Bx\|_1$, where $B \in \mathbb{R}^{n\times n}$ is a two dimensional Haar wavelet transform and $\lambda > 0$. Since $B$ is orthogonal,

we can reformulate the problem $\min_x f(x) + \varphi(x)$ into

$$\min_y \sum_{i=1}^{n} \log\left((AB^T y - b)_i^2 + 1\right) + \lambda \|y\|_1,$$

where $y := Bx$. The function $f$ is not convex, but $\nabla f$ is Lipschitz continuous. Furthermore, we expect a solution to this problem to approximately fulfill $AB^T y^* = b$, so $f$ is strongly convex in a neighbourhood of the solution if $A$ as full range. This means that our convergence theory applies here and we again get the convergence of the complete sequence of iterates to a stationary point.

We follow the test setting in [9], see also [29, 60], to restore a $256 \times 256$ test image, hence $n = 256^2 = 65536$. The mapping $A$ is a Gaussian blur operator of size $9 \times 9$ and with standard deviation 4 and $B$ is the two dimensional discrete Haar wavelet of level 4. Furthermore, we choose $\lambda = 10^{-4}$. The noisy blurred image $b$ is created from the original cameraman image by applying $A$ and adding Student's $t$-noise with degree of freedom 1 and rescaled by $10^{-3}$, and we start with $y^0 = b$.

For our analysis, we solve the image restoration with RPQN and QGPN with limited memory SR1-updates and a memory of 2 (which, again, behaved best in our tests), SNF and SpaRSA. Details on the methods are given in Section 7.1. Note that we do not apply FISTA to this problem since this solver is designed for convex problems.

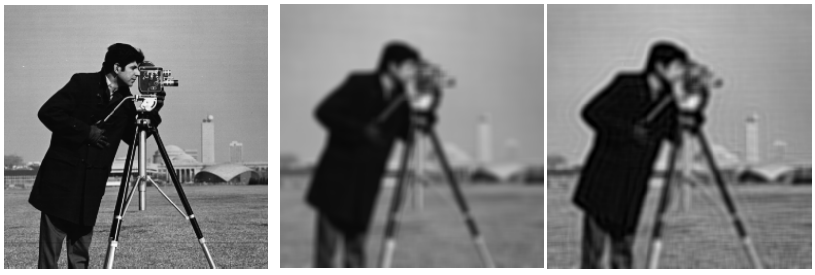
As before, using the same rules, we sometimes skip the limited memory updates. However, even though the problem is nonconvex and one can therefore expect that this case occurs more frequently, our experiments reveal that there is a maximum of one or two skipped updates per run of RPQN.

Here, we do not compute $\psi^*$ as the optimal value of the objective function, but as the function value of the original image (which are not the same in this case). For that reason, we terminate the methods if $\psi(x^k) \leq \psi(x^*)$ holds for an iterate $x^k$. The results, again averaged over 10 runs, are shown in Figure 3 (b). For the first iterations, all methods show similar performance and there are only minor differences. At some point, however, RQPN and shortly after QGPN instantly satisfy the termination criterion, whereas SpaRSA performs several more iterations until this goal is reached. Note that the performance of SNF is not satisfactory in this example and not shown in Figure 3 (b). In the nonconvex setting, this might be due to the structure, where semismooth iterations reducing $\|r(x^k)\|$ but probably increasing $\psi(x^k)$ and proximal gradient iterations, which decrease $\psi(x^k)$ but probably increase $\|r(x^k)\|$ are expected to alternate. We report some of the resulting data in Table 2.

Looking at the performance in Figure 3 (b), we also display the resulting images of the tested methods after a computation time of 12 seconds (and not using the above termination criterion) in Figure 4. It can be observed that RPQN and QGPN restore the image relatively well, while the result of SpaRSA is also satisfactory, but SNF is clearly outperformed.

| method | iter | Newton-iter | succ. iter | sub-iter | function eval | proximity eval | matrix-vector products |
|---|---|---|---|---|---|---|---|
| RPQN | 890 | - | 866 | 1790 | 891 | 4448 | 1790 |
| QGPN | 1101 | 1098 | - | 1175 | 1113 | 2354 | 2215 |
| SNF | 183 | 91 | - | 1189 | 784 | 408 | 3855 |
| SpaRSA | 1089 | - | - | 1964 | 1965 | 1964 | 3930 |

Table 2: Numerical data for the image restoration example in Section 7.3.



(a) Original Image (b) Noisy Image (c) SNF

(d) RPQN (e) QGPN (f) SpaRSA

Figure 4: Nonconvex image restoration from Section 7.3: Original and noisy image and recovered images using the stated algorithms and terminated after a computation time of 12 seconds.

## 8 Final Remarks

In this paper, we proposed a proximal quasi-Newton method with a regularization technique for a globalization, and presented the corresponding global convergence theory. After that we described a very efficient method for the computation of the occurring proximity operators using compact representations of limited memory quasi-Newton matrices. The numerical results show that the regularized method in combination with the efficient proximity operator computation accelerates the performance and outperforms both some standard first-order and some second-order methods.

Since our focus was on the limited memory quasi-Newton approach, we only presented a global convergence theory. A future approach is therefore to develop local convergence results under appropriate assumptions including a convergence assumption on the matrices $B_k$.

Furthermore, a main issue is the assumption that the convex function $\varphi$ is real-valued, and this fact is exploited in several steps of the current analysis. In the authors' opinion, the deduced algorithm should perform well also for problems with extended-valued functions $\varphi$. Thus, a main task of future research is the investigation of the convergence theory for this class of functions.

Finally, the computation of the variable metric proximity operators can be investigated. Many authors [8,24,58,67] provide convergence results for inexact solutions of this problem in

the setting of their proposed methods. Although our experiments reach very high accuracies in solving the subproblems within a very few steps, an improvement of the presented method could be to connect it to some of these criteria.

## References

[1] A.Y. Aravkin, M.P. Friedlander, F.J. Herrmann and T. Van Leeuwen, Robust inversion, dimensionality reduction, and randomized sampling, *Math. Program.* 134 (2012) 101–125.

[2] A.Y. Aravkin, R. Baraldi and D. Orban, A proximal quasi-Newton trust-region method for nonsmooth regularized optimization, *SIAM J. Optim.* 32 (2022) 900–929.

[3] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books in Mathematics, Springer International Publishing, second edition 2017.

[4] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (2009) 183–202

[5] S. Becker and J. Fadili, A quasi-Newton proximal splitting method, in: *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J. Burges, L. Bottou and K.Q. Weinberger (eds.), 2012, pp. 2618–2626.

[6] S. Becker, J. Fadili and P. Ochs, On quasi-Newton forward-backward splitting: Proximal calculus and convergence, *SIAM J. Optim.* 29 (2019) 2445–2481.

[7] S. Bonettini, I. Loris, F. Porta and M. Prato, Variable metric inexact line-search-based methods for nonsmooth optimization, *SIAM J. Optim.* 26 (2016) 891–921.

[8] S. Bonettini, I. Loris, F. Porta, Federica, M. Prato and S. Rebegoldi, On the convergence of a linesearch based proximal-gradient method for nonconvex optimization, *Inverse Problems* 33 (2017): 055005.

[9] R.I. Boţ, E.R. Csetnek S.C. László, An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions, *EURO J. Comput. Optim.* 4 (2016) 3–25.

[10] R.H. Byrd, G.M. Chin, J. Nocedal and F. Oztoprak, A family of second-order methods for convex $\ell_1$-regularized optimization, *Math. Program.* 159 (2016) 435–467.

[11] R.H. Byrd, J. Nocedal and F. Oztoprak, An inexact successive quadratic approximation method for L-1 regularized optimization, *Math. Program.* 157 (2016) 375–396.

[12] R.H. Byrd, J. Nocedal and R.B. Schnabel, Representations of quasi-Newton matrices and their use in limited memory methods, *Math. Program.* 63 (1994) 129–156.

[13] D.-Q. Chen, Y. Zhou and L.-J. Song, Fixed point algorithm based on adapted metric method for convex minimization problem with application to image deblurring, *Adv. Comput. Math.* 42 (2016) 1287–1310.

[14] P. Chen, J. Huang and X. Zhang, A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration, *Inverse Problems* 29 (2013): 025011.

[15] Z. Chen, A. Milzarek and Z. Wen, A trust-region method for nonsmooth nonconvex optimization, *J. Comput. Math.* 41 (2022) 683–716.

[16] F.H. Clarke, Generalized gradients and applications, *Trans. Amer. Math. Soc.* 205 (1975) 247–262.

[17] P.L. Combettes and V.R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Model. Simul.* 4 (2005) 1168–1200.

[18] A. Conn, N. Gould and P. Toint, *Trust Region Methods*, MPS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, 2000.

[19] M.A. Figueiredo, R.D. Nowak and S.J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J. Selected Topics in Signal Process.* 1 (2007) 586–597.

[20] R. Fletcher, A model algorithm for composite nondifferentiable optimization problems, in: *Nondifferential and Variational Techniques in Optimization*, D.C. Sorensen and R.J.-B. Wets (eds.), Springer, 1982, pp. 67–76.

[21] K. Fountoulakis and R. Tappenden, A flexible coordinate descent method, *Comput. Optim. Appl.* 70 (2018) 351–394.

[22] M.P. Friedlander and G. Goh, Efficient evaluation of scaled proximal operators, *Electron. Trans. Numer. Anal.* 46 (2017) 1–22.

[23] M. Fukushima and H. Mine, A generalized proximal point algorithm for certain nonconvex minimization problems, *nternat. J. Systems Sci.* 12 (1981) 989–1000.

[24] H. Ghanbari and K. Scheinberg, Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates, *Comput. Optim. Appl.* 69 (2018) 597–627.

[25] R. Griesse and D. A. Lorenz, A semismooth Newton method for Tikhonov functionals with sparsity constraints, *Inverse Problems* 24 (2008): 035007.

[26] B. Gu, D. Wang, Z. Huo and H. Huang, Inexact proximal gradient methods for nonconvex and non-smooth optimization, in: *32. AAAI Conference on Artificial Intelligence*, 2018, pp. 3093–3100.

[27] E.T. Hale, W. Yin and Y. Zhang, Fixed-point continuation for $\ell_1$-minimization: Methodology and convergence, *SIAM J. Optim.* 19 (2008) 1107–1130.

[28] K. Jiang, D. Sun and K.-C. Toh, An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP, *SIAM J. Optim.* 22 (2012) 1042–1064.

[29] C. Kanzow and T. Lechner, Globalized inexact proximal Newton-type methods for nonconvex composite functions, *Comput. Optim. Appl.* 78 (2021) 377–410.

[30] IMRO: A proximal quasi-newton method for solving $\ell_1$-regularized least squares problems, *SIAM J. Optim.* 27 (2017) 583–615.

[31] D. Kim, S. Sra and I.S. Dhillon, A scalable trust-region algorithm with application to mixed-norm regression, in: *27th International Conference on Machine Learning*, J. Fürnkranz and T. Joachims (eds.), Madison, WI, Omnipress, 2010, pp. 519–526.

[32] S.-J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, An interior-point method for large-scale $\ell_1$-regularized least squares, *IEEE J. Selected Topics in Signal Process.* 1 (2007) 606–617.

[33] K. Koh, S.-J. Kim and S. Boyd, An interior-point method for large-scale $\ell_1$-regularized logistic regression, *J. Mach. Learn. Res.* 8 (2007) 1519–1555.

[34] C.-P. Lee, C H. Lim and S.J. Wright, A distributed quasi-Newton algorithm for empirical risk minimization with nonsmooth regularization, in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2018, pp. 1646–1655.

[35] C.-P. Lee and S.J. Wright, Inexact successive quadratic approximation for regularized optimization, *Comput. Optim. Appl.* 72 (2019) 641–674.

[36] J.D. Lee, Y. Sun and M.A. Saunders, Proximal Newton-type methods for minimizing composite functions, *SIAM J. Optim.* 24 (2014) 1420–1443.

[37] D.-H. Li, M. Fukushima, L. Qi and N. Yamashita, Regularized Newton methods for convex minimization problems with singular solutions, *Comput. Optim. Appl.* 28 (2004) 131–147.

[38] G. Li and T.K. Pong, Calculus of the exponent of Kurdyka–Lojasiewicz inequality and its applications to linear convergence of first-order methods, *Found. Comput. Math.* 18 (2018) 1199–1232.

[39] X. Li, D. Sun and K.-C. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems, *SIAM J. Optim.* 28 (2018) 433–458.

[40] D.C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program.* 45 (1989) 503–528.

[41] R. Liu, S. Pan, Y. Wu and X. Yang, An inexact regularized proximal Newton method for nonconvex and nonsmooth optimization, *Comput. Optim. Appl.* 88 (2024) 603–641.

[42] Z.-Q. Luo and P. Tseng, Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem, *SIAM J. Optim.* 2 (1992) 43–54.

[43] Z.-Q. Luo and P. Tseng, Error bounds and convergence analysis of feasible descent methods: a general approach, *Ann. Oper. Res.* 46 (1993) 157–178.

[44] L. Meier, S. Van De Geer and P. B 端 hlmann, The group lasso for logistic regression, *J. R. Stat. Soc. Ser. B. Stat. Methodol.* 70 (2008) 53–71.

[45] A. Milzarek, *Numerical Methods and Second Order Theory for Nonsmooth Problems*, PhD thesis, Technical University of Munich, 2016.

[46] A. Milzarek and M. Ulbrich, A semismooth Newton method with multidimensional filter globalization for $\ell_1$-optimization, *SIAM J. Optim.* 24 (2014) 298–333.

[47] B.S. Mordukhovich, X. Yuan, S. Zeng and J. Zhang, A globally convergent proximal Newton-type method in nonsmooth convex optimization, *Math. Program.* 198 (2023) 899–936.

[48] J.-J. Moreau, Proximité et dualité dans un espace Hilbertien, *Bull. Soc. Math. France* 93 (1965) 273–299.

[49] P.Q. Muoi, D.N. Hào, P. Maass and M. Pidcock, Semismooth Newton and quasi-Newton methods in weighted $\ell_1$-regularization, *J. Inverse Ill-Posed Probl.* 21 (2013) 665–693.

[50] S. Nakayama, Y. Narushima and H. Yabe, Inexact proximal memoryless quasi-Newton methods based on the Broyden family for minimizing composite functions, *Comput. Optim. Appl.* 79 (2021) 127–154.

[51] Y. Nesterov, Gradient methods for minimizing composite functions, *Math. Program.* 140 (2013) 125–161.

[52] J. Nocedal, Updating quasi-Newton matrices with limited storage, *Math. Comp.* 35 (1980) 773–782.

[53] P. Ochs and T. Pock, Adaptive FISTA for nonconvex optimization, *SIAM J. Optim.* 29 (2019) 2482–2503.

[54] B. Pötzl, A. Schiela and P. Jaap, Inexact proximal Newton methods in Hilbert spaces, arXiv preprint arXiv:2204.12168 (2022).

[55] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Math. Program.* 58 (1993) 353–367.

[56] L. Qi and J. Sun, A trust region algorithm for minimization of locally Lipschitzian functions, *Math. Program.* 66 (1994) 25–43.

[57] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 2015.

[58] K. Scheinberg and X. Tang, Practical inexact proximal quasi-Newton method with global complexity analysis, *Math. Program.* 160 (2016) 495–529.

[59] D. Steck and C. Kanzow, Regularization of limited memory quasi-Newton methods for large-scale nonconvex minimization, *Math. Program. Comput.* 15 (2023) 417–444.

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

[61] A. Themelis, L. Stella and P. Patrinos, Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms, *SIAM J. Optim.* 28 (2018) 2274–2303.

[62] P. Tseng and S. Yun, A coordinate gradient descent method for nonsmooth separable mini- mization, *Math. Program.* 117 (2009) 387–423.

[63] K. Ueda and N. Yamashita, Convergence properties of the regularized Newton method for the unconstrained nonconvex optimization, *Appl. Math. Optim.* 62 (2010) 27–46.

[64] K. Ueda and N. Yamashita, A regularized Newton method without line search for unconstrained optimization, *Comput. Optim. Appl.* 59 (2014) 321–351.

[65] 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) 1832–1857.

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

[67] M.-C. Yue, Z. Zhou and A.M.-C. So, A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo-Tseng error bound property, *Math. Program.* 174 (2019) 327–358.

[68] Z. Zhou and A.M.-C. So, A unified approach to error bounds for structured convex optimization problems, *Math. Program.* 165 (2017) 689–728.

---

*Manuscript received 28 July 2023*
*revised 17 October 2023*
*accepted for publication 31 October 2023*

Christian Kanzow
University of Würzburg, Institute of Mathematics
Emil-Fischer-Str. 30, 97074 Würzburg, Germany
E-mail address: christian.kanzow@uni-wuerzburg.de

Theresa Lechner
University of Würzburg, Institute of Mathematics
Emil-Fischer-Str. 30, 97074 Würzburg, Germany
E-mail address: theresa.lechner@uni-wuerzburg.de