Computational and Applied Mathematics manuscript No.
(will be inserted by the editor)

Efficient Proximal Subproblem Solvers for a Nonsmooth Trust-Region Method

Robert J. Baraldi · Drew P. Kouri

Received: date / Accepted: date

Abstract In [R. J. Baraldi and D. P. Kouri, *Mathematical Programming*, (2022), pp. 1-40], we introduced an inexact trust-region algorithm for minimizing the sum of a smooth nonconvex and nonsmooth convex function. The principle expense of this method is in computing a trial iterate that satisfies the so-called fraction of Cauchy decrease condition—a bound that ensures the trial iterate produces sufficient decrease of the subproblem model. In this paper, we expound on various proximal trust-region subproblem solvers that generalize traditional trust-region methods for smooth unconstrained and convex-constrained problems. We introduce a simplified spectral proximal gradient solver, a truncated nonlinear conjugate gradient solver, and a dogleg method. We compare algorithm performance on examples from data science and PDE-constrained optimization.

Keywords Nonsmooth Optimization · Nonlinear Programming · Trust Regions · Large-Scale Optimization · Proximal Newton's Method

---

This research was sponsored, in part, by the Department of Energy Office of Science under the Early Career Research Program and the U.S. Air Force Office of Scientific Research. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Drew P. Kouri
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87125, USA
E-mail: dpkouri@sandia.gov
ORCID: 0000-0002-7079-3195

Robert J. Baraldi (Corresponding Author)
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87125, USA
E-mail: rjbaral@sandia.gov
ORCID: 0000-0003-3699-6770

Mathematics Subject Classification (2020) 49M15 · 49M37 · 65K05 · 65K10 · 90C06 · 90C30

1 Introduction

In [4], we developed a trust-region method for the nonsmooth optimization problem

$$\min_{x \in X} \; f(x) + \phi(x), \tag{1}$$

where X is a Hilbert space, $f : X \to \mathbb{R}$ is Fréchet differentiable with Lipschitz continuous gradient, and $\phi : X \to (-\infty, +\infty]$ is proper, closed and convex. The method introduced in [4] permits and systematically controls inexactness in the evaluations of f and its gradient ∇f, while guaranteeing convergence. This enables the numerical solution of infinite-dimensional optimization problems, where finite-dimensional approximations are indispensable for evaluating f and ∇f.

Inexactness notwithstanding, typical trust-region methods measure progress using a Cauchy point (CP) or, more generally, a fraction of Cauchy decrease (FCD) condition [18,37,42,53]. For smooth unconstrained problems, the CP is the minimizer of a quadratic model in the steepest descent direction. When simple constraints are present, the CP is any point along the projected gradient path that produces sufficient decrease of the model [42,53]. In [4], we generalized the CP to a point along the proximal gradient path and computed it using a bidirectional proximal search, cf. [4, Alg. 2]. In this paper, we develop various trust-region subproblem solvers that improve upon the CP and are guaranteed to satisfy the FCD condition, thereby ensuring convergence of the trust-region algorithm [4, Alg. 1]. Moreover, our subproblem solvers ensure rapid superlinear, even quadratic, convergence of the trust-region algorithm when the problem data in (1) permits [5].

Since the inception of trust-region methods, numerous subproblem solvers have been proposed, primarily for smooth problems. Early methods were so-called *dogleg* approaches because they employ a piecewise linear interpolation between the CP and unconstrained Newton point to guarantee fraction of Cauchy decrease; cf. Powell [44,45]. Powell's dogleg method was extended in [21] to a double dogleg path by adding an additional piecewise linear segment that biases the Newton point, yielding improved local convergence. Dogleg methods are computationally simple but produce potentially poor trial iterates near the trust-region radius. To overcome this, Moré and Sorensen computed trial iterates by solving the reformulated subproblem first-order optimality conditions with Newton's method [41]. One could similarly solve the subproblem using Gaussian quadrature [28]. Around the same time as [41], Steihaug [50] and Toint [52] introduced the truncated conjugate gradient (CG) method, which approximately solves the subproblem using CG modified with stopping conditions that account for negative curvature and the trust-region constraint. Truncated CG has also been used within trust-region methods for solving various constrained optimization problems [30,31,37]. Motivated by truncated CG, the authors in [29], proposed solving the subproblem using a truncated Lanczos method. More recently, [32,39] employed the spectral projected gradient method [8] to compute a trial iterate for smooth unconstrained and convex-constrained problems.

The trust-region subproblem used in [4, Alg. 1] is

$$\min_{x\in X}\ \{m_k(x) \coloneqq f_k(x) + \phi(x)\} \qquad \text{subject to} \qquad \|x - x_k\| \le \Delta_k, \tag{2}$$

where $x_k \in X$ is the current iterate, f_k is a local approximation of f around x_k, and $\Delta_k > 0$ is the current trust-region radius. The presence of nonsmooth ϕ in (2) renders most of the aforementioned methods irrelevant. To rectify this, we introduce extensions of these classical methods that verifiably produce trial iterates satisfying the FCD condition. We establish three main solvers: 1) a simplified spectral proximal gradient (SPG) method; 2) a nonsmooth truncated CG method; and 3) a nonsmooth dogleg method. Our SPG method streamlines the algorithm proposed in [4, Alg. 5] by using a simplified spectral CP and handling the trust-region constraint separately from the proximity operator computation. These modifications typically result in fewer evaluations of the proximity operator. Our truncated CG approach is based on nonlinear CG with modifications that account for the nonsmooth term as well as the trust-region constraint. For our dogleg framework, we compute the Newton point using damped semismooth Newton, which requires the application of a generalized Jacobian of the proximity operator. Fortunately, the proximity operators for numerous ϕ are semismooth [10]. In the appendix, we include a specialized orthant-based subproblem solver for L^1-regularized problems based on [13].

We organize the paper as follows. Section 2 introduces the notation and problem assumptions. Section 3 reviews the trust-region algorithm from [4] and highlights its basic functionality. Section 4 discusses global and local convergence of the algorithm. Section 5 details the subproblem solvers, and Section 6 compares their performance on six numerical examples arising from data science and optimization problems constrained by partial differential equations (PDEs).

2 Notation and Problem Assumptions

Let X be a Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\|$, and let $\mathcal{L}(X)$ denote the space of continuous linear operators that map X into itself. Recall that $\mathcal{L}(X)$ is a Banach space endowed with the usual operator norm

$$\|B\| = \sup\{\|Bx\| \mid \|x\| \le 1\} \qquad \forall\, B \in \mathcal{L}(X).$$

To simplify the presentation, we identify the topological dual space X^* with X via Riesz representation. Following standard convex analysis notation, we denote the subdifferential of a proper, closed and convex function $\psi : X \to (-\infty, \infty]$ by

$$\partial\psi(x) \coloneqq \{\eta \in X \mid \psi(y) \ge \psi(x) + \langle \eta, y - x\rangle \quad \forall\, y \in X\}.$$

and the effective domains of ψ and $\partial\psi$ by

$$\operatorname{dom}\psi \coloneqq \{x \in X \mid \psi(x) < +\infty\} \qquad \text{and} \qquad \operatorname{dom}\partial\psi \coloneqq \{x \in X \mid \partial\psi(x) \ne \emptyset\},$$

respectively. Furthermore, the proximity operator of ψ is

$$\operatorname{Prox}_{r\psi}(x) \coloneqq \operatorname*{arg\,min}_{y\in X}\ \{\psi(y) + \tfrac{1}{2r}\,\|y - x\|^2\}, \tag{3}$$

for $r > 0$. When $\psi = \iota_{\mathcal{C}}$ is the indicator function of a nonempty, closed and convex set $\mathcal{C} \subset X$ (i.e., $\iota_{\mathcal{C}}(x) = 0$ if $x \in \mathcal{C}$ and $\iota_{\mathcal{C}}(x) = +\infty$ if $x \notin \mathcal{C}$), $\mathrm{Prox}_{r\psi}(x)$ is the projection of x onto $\mathcal{C}$. In the subsequent sections, we make repeated use of the proximity operator's firm nonexpansivity [6, Prop. 12.27]. For other useful properties of the proximity operator, see [4, Sec. 2.2];

The convergence theory in [4] requires the following standard assumptions on the problem data in (1).

Assumption 1 (Problem Data) *The components of the objective function*

$$F(x) \coloneqq f(x) + \phi(x)$$

in (1) *satisfy the following conditions.*

1. *The function $\phi : X \to (-\infty, +\infty]$ is proper, closed and convex.*
2. *The function $f : X \to \mathbb{R}$ is L-smooth on* $\mathrm{dom}\,\phi$*. That is, f is Fréchet differentiable and its gradient ∇f is Lipschitz continuous with modulus $L > 0$ on an open set $U \subseteq X$ containing* $\mathrm{dom}\,\phi$*.*
3. *The objective function F is bounded below, i.e., there exists $\kappa_{\mathrm{lb}} \in \mathbb{R}$ such that $F(x) \geq \kappa_{\mathrm{lb}}$ for all $x \in X$.*

Recall that if $\bar{x} \in X$ is a local minimizer for (1), then it satisfies

$$-\nabla f(\bar{x}) \in \partial\phi(\bar{x}) \qquad \Longleftrightarrow \qquad \bar{x} = \mathrm{Prox}_{r\phi}(\bar{x} - r\nabla f(\bar{x}))$$

for arbitrary, fixed $r > 0$. The second condition above motivates a natural algorithmic stopping condition. Commonly, algorithms for (1) will stop iterating if the current iterate $x \in X$ satisfies

$$\tfrac{1}{r}\, \|x - \mathrm{Prox}_{r\phi}(x - r\nabla f(x))\| \leq \tau,$$

for a user-specified tolerance $\tau > 0$ and fixed $r > 0$. For use in later sections, we define the functions $G : X \times X \times [0, \infty) \to X$, $G_f : X \times [0, \infty) \to X$, $H : X \times X \times [0, \infty) \to \mathbb{R}$ and $h : X \times [0, \infty) \to \mathbb{R}$ by

$$\begin{aligned} &G(x, g, r) \coloneqq \tfrac{1}{r}(x - \mathrm{Prox}_{r\phi}(x - rg)), \quad G_f(x, r) \coloneqq G(x, \nabla f(x), r) \\ &H(x, g, r) \coloneqq \|G(x, g, r)\| \quad \text{and} \quad h(x, r) \coloneqq \|G_f(x, r)\|, \end{aligned} \tag{4}$$

respectively. The next proposition catalogues important properties of G and H.

Proposition 1 (Properties of G and H)

a: *For fixed x, $g \in X$, $r \mapsto rH(x, g, r)$ is nondecreasing on $(0, \infty)$. In particular, if $r \geq t > 0$, then $rH(x, g, r) \geq tH(x, g, t)$. Moreover, this inequality is strict if $rG(x, g, r) \neq tG(x, g, t)$.*

b: *For fixed x, $g \in X$, $r \mapsto H(x, g, r)$ is nonincreasing for $r > 0$.*

c: *For fixed x, $g \in X$ and $r > 0$, the following inequality holds*

$$-r\,\langle g, G(x, g, r)\rangle + \phi(x - rG(x, g, r)) - \phi(x) \leq -rH(x, g, r)^2. \tag{5}$$

d: *The maps $(x, g, r) \mapsto G(x, g, r)$ and $(x, g, r) \mapsto H(x, g, r)$ are continuous on $X \times X \times (0, \infty)$.*

e: *For fixed* $r > 0$, $(x, g) \mapsto H(x, g, r)$ *satisfies*

$$|H(x,g,r) - H(x',g',r)| \le \frac{1}{r}\|x - x'\| + \|g - g'\| \quad \forall\, x,\, x',\, g,\, g' \in X. \tag{6}$$

In particular, $(x, g) \mapsto H(x, g, r)$ *is Lipschitz continuous.*

Proof Parts a, b and c are direct consequences of [4, Lem. 2 & 3] with $d = -g$. For part d, we recall that $r \mapsto \mathrm{Prox}_{r\phi}(y)$ is continuous for fixed $y \in X$ [4, Lem. 3] and $y \mapsto \mathrm{Prox}_{r\phi}(y)$ is Lipschitz continuous with unit modulus for fixed $r \in (0, \infty)$. Now, suppose $\{(y_n, r_n)\} \subset X \times (0, \infty)$ with $y_n \to y$ and $r_n \to r > 0$. By Lipschitz continuity, we have that

$$\|\mathrm{Prox}_{r_n\phi}(y_n) - \mathrm{Prox}_{r\phi}(y)\| \le \|y_n - y\| + \|\mathrm{Prox}_{r_n\phi}(y) - \mathrm{Prox}_{r\phi}(y)\|\,.$$

Consequently, $(y, r) \mapsto \mathrm{Prox}_{r\phi}(y)$ is continuous. Hence, the composition of this map with the continuous map $(x, g, r) \mapsto (x - rg, r)$ is also continuous. Part e follows from the firm nonexpansivity of the proximity operator. □

3 Trust-Region Algorithm

To facilitate subproblem solver development, we choose f_k in (2) to be the quadratic model

$$f_k(x) := \frac{1}{2}\langle B_k(x - x_k), x - x_k\rangle + \langle g_k, x - x_k\rangle\,, \tag{7}$$

where $B_k \in \mathcal{L}(X)$ is self adjoint and $g_k \in X$ is an approximation of $\nabla f(x_k)$. The operator B_k encapsulates the curvature of f at x_k and is often the Hessian $\nabla^2 f(x_k)$ or a secant approximation thereof. At the k-th iteration of the trust-region algorithm introduced in [4, Alg. 1], one computes a trial iterate x_k^+ that satisfies two conditions: there exists positive constants κ_{rad}, $\kappa_{\mathrm{fcd}} > 0$, independent of k, and a positive parameter $t_k > 0$ for which *(i)* the trust-region constraint

$$\left\|x_k^+ - x_k\right\| \le \kappa_{\mathrm{rad}}\Delta_k \tag{8a}$$

holds, and *(ii)* the FCD

$$m_k(x_k) - m_k(x_k^+) \ge \kappa_{\mathrm{fcd}} h_k \min\left\{\frac{h_k}{1 + \|B_k\|}, \Delta_k\right\} \tag{8b}$$

is satisfied, where

$$h_k := H(x_k, g_k, t_k) \tag{9}$$

with H defined in (4). Note that (8b) ensures that $x_k^+ \in \mathrm{dom}\,\phi_k$ since the left-hand side would be $-\infty$ otherwise. Additionally, note that (8b) is a slight generalization of [4, Eq. (12b)], where t_k is a constant independent of the iteration number k. It is common to choose t_k to be the computed CP step length as is done in the linear-constrained trust-region method of [37].

Given a trial iterate x_k^+ that satisfies (8), the trust-region algorithm accepts or rejects x_k^+ based on the ratio of actual and predicted reduction

$$\rho_k^* := \frac{\mathrm{ared}_k}{\mathrm{pred}_k},$$

where

$$\text{ared}_k \coloneqq F(x_k) - F(x_k^+) \qquad \text{and} \qquad \text{pred}_k \coloneqq m_k(x_k) - m_k(x_k^+).$$

Here, ared_k is the actual reduction of the objective function F achieved by x_k^+ relative to x_k and pred_k is the reduction predicted by the model m_k. In many practical applications, the objective function F cannot be computed accurately [19,27,34,35], necessitating the replacement of ared_k in ρ_k^* with an approximation denoted cred_k—the *computed reduction*. Algorithmically, we decide whether or not to accept x_k^+ based on the ratio of computed and predicted reduction

$$\rho_k \coloneqq \frac{\text{cred}_k}{\text{pred}_k}. \tag{10}$$

We set $x_{k+1} = x_k^+$ if $\rho_k \geq \eta_1$ and $x_{k+1} = x_k$ otherwise. The trust-region algorithm then increases the radius Δ_k if $\rho_k \geq \eta_2$ and reduces Δ_k if $\rho_k < \eta_1$. The algorithmic parameters $0 < \eta_1 < \eta_2 < 1$ are user-specified with common values $\eta_1 = 10^{-4}$ and $\eta_2 = 0.75$.

To ensure cred_k is a sufficiently accurate approximation of ared_k, we require the following assumption.

Assumption 2 (Inexact Objective Function) *The accuracy of the computed reduction* cred_k *can be refined to satisfy the condition: there exists a constant* $\kappa_{\text{obj}} \geq 0$*, independent of* k*, such that*

$$|\text{ared}_k - \text{cred}_k| \leq \kappa_{\text{obj}} \left[\eta \min\{\text{pred}_k, \theta_k\}\right]^\zeta \quad \forall\, k, \tag{11}$$

where ζ*,* η*, and* θ_k *are (user-specified) positive real numbers that satisfy*

$$\zeta > 1, \qquad 0 < \eta < \min\{\eta_1, (1-\eta_2)\}, \qquad \textit{and} \qquad \lim_{k\to+\infty} \theta_k = 0.$$

Here, ζ *and* η *are independent of* k*.*

Condition (11) was first used in [34], where it was motivated by [55, Sec. 5.3.3]. Recent work [7,51] has developed trust-region methods for general noisy objective functions that require an explicit bound on the noise (cf. [7, Eq. (2.16)] and [51, Eq. (3)]). This requirement is often impossible to satisfy in, e.g., infinite-dimensional problems, where the left-hand side of (11) may only be bounded by an error indicator that depends on uncomputable constants like continuity, embedding, or inf-sup constants in PDE applications [3,12].

Assumption 2 enables us to inexactly evaluate the objective function F. Moreover, since $\theta_k \to 0$ as $k \to +\infty$, we have that $\theta_k \leq \kappa_{\text{obj}}^{-1/(\zeta-1)}$ for sufficiently large k and

$$|\text{ared}_k - \text{cred}_k| \leq \kappa_{\text{obj}}[\eta \min\{\text{pred}_k, \theta_k\}]\theta_k^{\zeta-1} \leq \eta \min\{\text{pred}_k, \theta_k\}. \tag{12}$$

The consequence of (12) on the accuracy of ρ_k is summarized in the next lemma.

Lemma 1 (Lemma 6 in [4]) *If Assumption 2 holds, then there exists a positive integer* K_η *satisfying*

$$\rho_k^* = \frac{\text{ared}_k}{\text{pred}_k} \in [\rho_k - \eta, \rho_k + \eta] \qquad \forall\, k \geq K_\eta. \tag{13}$$

Lemma 1 ensures that successful steps produce sufficient decrease in F as demonstrated in the following corollary.

Corollary 1 (Corollary 2 in [4]) *Let Assumption 2 hold and suppose x_k^+ is a trial iterate that satisfies* (8b) *with $\rho_k \geq \eta_1$, then*

$$\mathrm{cred}_k \geq \eta_1 \kappa_{\mathrm{fcd}} h_k \min\left\{\frac{h_k}{1+\|B_k\|}, \Delta_k\right\}.$$

Moreover, if $k \geq K_\eta$ where K_η is defined in Lemma 1, then

$$\mathrm{ared}_k \geq (\eta_1 - \eta) \kappa_{\mathrm{fcd}} h_k \min\left\{\frac{h_k}{1+\|B_k\|}, \Delta_k\right\}.$$

As with the computed reduction, the gradient of f often cannot be evaluated exactly in practice [19,27,33,35]. Instead, we require that the approximate gradient satisfies the following assumption.

Assumption 3 (Subproblem Model) *The accuracy of the model gradient g_k can be refined to satisfy the condition: there exists a constant $\kappa_{\mathrm{grad}} \geq 0$, independent of k, such that*

$$\|g_k - \nabla f(x_k)\| \leq \kappa_{\mathrm{grad}} \min\{h_k, \Delta_k\} \quad \forall\, k. \tag{14}$$

Again, Assumption 3 differs from the general noisy objective function setting employed in [7,51] as it enables the use of error indicators that generally depend on uncomputable constants. With Assumptions 2 and 3, we can now state the trust-region algorithm for solving (1), listed in Algorithm 1.

Algorithm 1 Inexact Nonsmooth Trust-Region Algorithm

Require: Initial guess $x_0 \in \mathrm{dom}\,\phi$, initial radius $\Delta_0 > 0$, $0 < \eta_1 < \eta_2 < 1$, and $0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3$

for $k = 0, 1, 2, \ldots$ **do**
  Gradient Approximation: Compute g_k that satisfies Assumption 3
  Model Selection: Choose self-adjoint $B_k \in \mathcal{L}(X)$ and build m_k using (7)
  Step Computation: Compute $x_k^+ \in X$ that satisfies (8)
  Computed Reduction: Compute cred_k that satisfies Assumption 2
  Step Acceptance and Radius Update: Compute ρ_k as in (10)
  if $\rho_k < \eta_1$ **then**
    $x_{k+1} \leftarrow x_k$
    $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$
  else
    $x_{k+1} \leftarrow x_k^+$
    if $\rho_k \in [\eta_1, \eta_2)$ **then**
      $\Delta_{k+1} \in [\gamma_2 \Delta_k, \Delta_k]$
    else
      $\Delta_{k+1} \in [\Delta_k, \gamma_3 \Delta_k]$
    end if
  end if
end for

4 Convergence Analysis

The global convergence analysis of Algorithm 1 is essentially the same as in [4], despite the more general h_k definition. As such, we state the basic convergence results without proof unless significant modification is required.

Theorem 1 *Let $\{x_k\}$ be the sequence of iterates generated by Algorithm 1. If Assumptions 1, 2 and 3 hold and if*

$$\sum_{k=0}^{\infty}(1+\max_{i=1,\ldots,k}\|B_i\|)^{-1}=+\infty, \tag{15}$$

then

$$\liminf_{k\to\infty} h_k = 0 \qquad \text{and} \qquad \liminf_{k\to\infty} h(x_k,t_k)=0. \tag{16}$$

Proof The proof of this result is identical to that of [4, Th. 3].

Under mild additional assumptions, we can improve upon Theorem 1 to show that the limit of $h(x_k,t)$, not just the lower limit, is zero for all $t>0$.

Theorem 2 *Let the assumptions of Theorem 1 hold. In addition, suppose there exist $\kappa_{\mathrm{curv}}>0$ and $t_{\max}>0$ such that $\|B_k\|\le\kappa_{\mathrm{curv}}$ and $t_k\le t_{\max}$ for all k. Then,*

$$\lim_{k\to\infty} H(x_k,g_k,t)=0 \qquad \text{and} \qquad \lim_{k\to\infty} h(x_k,t)=0 \quad \forall\, t>0.$$

Proof By Theorem 1, the existence of $t_{\max}$ and Proposition 1, we have that

$$\liminf_{k\to\infty} H(x_k,g_k,t_{\max})=0 \qquad \text{and} \qquad \liminf_{k\to\infty} h(x_k,t_{\max})=0.$$

The result then follows from [5, Th. 1]. □

To derive a convergence rate for Algorithm 1, we require the following assumptions on the method used to generate the trial iterate x_k^+.

Assumption 4 (Subproblem Solver) *There exists $\mu\in(0,\frac{1}{2})$, independent of k, such that the trial iterate x_k^+ satisfies the decrease condition*

$$m_k(x_k^+)-m_k(x_k)\le\mu\left(\left\langle g_k,x_k^+-x_k\right\rangle+\phi(x_k^+)-\phi(x_k)\right)=:\mu Q_k, \tag{17}$$

the trust-region constraint (8a)*, and either*

$$H(x_k^+,\nabla f_k(x_k^+),t_k)\le\tau_k h_k \quad \text{or} \quad \left\|x_k^+-x_k\right\|=\kappa_{\mathrm{rad}}\Delta_k, \tag{18}$$

where $\{\tau_k\}\subset[0,\infty)$ is a bounded sequence of relative tolerances. Moreover, let $x_k^{\mathrm{n}}\in X$ be any point that satisfies the first condition in (18)*. If there exists x_k^{n} with $\|x_k^{\mathrm{n}}-x_k\|\le\kappa_{\mathrm{rad}}\Delta_k$, then x_k^+ also satisfies the first condition in* (18)*.*

The assumption that x_k^+ eventually behaves like an inexact Newton iterate x_k^n is common in the trust-region literature. For instance, similar conditions are used in [43, Th. 4.9] for smooth unconstrained optimization and [37] for smooth convex-constrained optimization.

Two of our subproblem solvers are iterative (cf. Algorithms 3 and 4), in which case x_k^+ is selected as the final element in a sequence of iterates, $\{x_{k,0}, x_{k,1}, \ldots, x_{k,n_k}\}$ with $x_{k,0} = x_k$ and $x_{k,n_k} = x_k^+$. For these solvers, we employ the iteration decrease condition

$$\begin{aligned} m_k(x_{k,\ell+1}) - &m_k(x_{k,\ell}) \\ &\le \mu\left(\langle \nabla f_k(x_{k,\ell}), x_{k,\ell+1} - x_{k,\ell}\rangle + \phi(x_{k,\ell+1}) - \phi(x_{k,\ell})\right) \end{aligned} \tag{19}$$

for $\ell = 0, 1, \ldots, n_k$, instead of (17). We further assume that the number of iterations is limited to `maxit`, i.e., $n_k \le$ `maxit` for $k = 1, 2, \ldots$. See [5] for the convergence analysis of iterative subproblem solvers. Under stronger assumptions than the preceding two theorems, the next result demonstrates that Algorithm 1 ultimately accepts every x_k^+, which eventually satisfies the first condition in (18).

Theorem 3 *Let the assumptions of Theorem 2 and Assumption 4 hold, and suppose there exists an open set $U_0 \subseteq X$ containing a stationary point $\bar{x}$ of (1) on which f is twice continuously Fréchet differentiable. Furthermore, suppose that $x_k \to \bar{x}$, $g_k = \nabla f(x_k)$, and B_k in (7) satisfies:*

1. *There exists $K_0 \in \mathbb{N}$ such that B_k is uniformly strongly monotone and bounded for $k \ge K_0$, i.e., there exist $m > 0$ and $\kappa_{\mathrm{curv}} > 0$ such that*
$$m\,\|s\|^2 \le \langle B_k s, s\rangle \quad \textit{and} \quad \|B_k\| \le \kappa_{\mathrm{curv}} \tag{20}$$
for all $s \in X$ and $k \ge K_0$; and
2. *The Dennis-Moré condition holds, i.e.,*
$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(\bar{x}))(x_k^+ - x_k)\|}{\|x_k^+ - x_k\|} = 0. \tag{21}$$

Then, there exists a positive integer K_1 such that $x_{k+1} = x_k^+$ and $\Delta_{k+1} \ge \Delta_k$ for all $k \ge K_1$.

Proof The main distinction between this result and [5, Th. 2] is the use of a quadratic model f_k whose Hessian satisfies the Dennis-Moré condition (21), in place of the gradient consistency condition M5 in [5].

We bound $|\rho_k^* - 1| = |(\mathrm{ared}_k - \mathrm{pred}_k)|/\mathrm{pred}_k$ and show that it converges to zero if $s_k := x_k^+ - x_k \to 0$. Suppose $s_k \to 0$. To bound the numerator, we note that the nonsmooth terms cancel. Therefore, Taylor's theorem applied to the twice continuously differentiable function $\sigma \mapsto f(x_k + \sigma s_k)$ (for k sufficiently large) ensures the existence of $\sigma_k \in [0,1]$ for which

$$\begin{aligned} |\mathrm{ared}_k - \mathrm{pred}_k| &= |f(x_k) - f(x_k^+) - f_k(x_k) + f_k(x_k^+)| \\ &= \tfrac{1}{2}\left|\left\langle (B_k - \nabla^2 f(x_k + \sigma_k s_k))s_k, s_k\right\rangle\right| \\ &\le \tfrac{1}{2}\left(\left\|(B_k - \nabla^2 f(\bar{x}))s_k\right\| + \left\|(\nabla^2 f(\bar{x}) - \nabla^2 f(x_k + \sigma_k s_k))s_k\right\|\right)\|s_k\|. \end{aligned}$$

Since $x_k \to \bar{x}$, $s_k \to 0$ and $\{\sigma_k\} \subset [0,1]$, we have that $x_k + \sigma_k s_k \to \bar{x}$. Therefore, the continuity of $\nabla^2 f$ and (21) ensure that

$$|\mathrm{ared}_k - \mathrm{pred}_k| \le o(\|s_k\|)\,\|s_k\| \qquad \text{as} \qquad \|s_k\| \to 0.$$

Moreover, the sufficient decrease condition (17) ensures that

$$\begin{aligned} \mu \mathcal{Q}_k &\ge m_k(x_k^+) - m_k(x_k) = \mathcal{Q}_k + \tfrac{1}{2}\langle B_k s_k, s_k\rangle \\ \iff \qquad & -(1-\mu)\mathcal{Q}_k \ge \tfrac{1}{2}\langle B_k s_k, s_k\rangle . \end{aligned}$$

Combining this with (20) yields

$$\mathrm{pred}_k \ge -\mu \mathcal{Q}_k = -\frac{\mu}{1-\mu}(1-\mu)\mathcal{Q}_k \ge \frac{\mu}{1-\mu}\frac{m}{2}\|s_k\|^2 =: \kappa_0 \|s_k\|^2 .$$

Combining the numerator and denominator bounds, we arrive at

$$|\rho_k^* - 1| \le o(\|s_k\|)\frac{\|s_k\|}{\kappa_0\|s_k\|^2} = o(1) \qquad \text{as} \qquad \|s_k\| \to 0.$$

Hence, if $\Delta_k \to 0$, then $\|s_k\| \to 0$ and consequently $|\rho_k^* - 1| \to 0$. Therefore, $\rho_k \ge \eta_2$ and $\Delta_{k+1} \ge \Delta_k$ for all k sufficiently large, which contradicts $\Delta_k \to 0$. The result then follows from [5, Cor. 2]. □

Our final result provides convergence rates for $\{x_k\}$ generated by Algorithm 1, when the trial iterates x_k^+ satisfy Assumption 4.

Theorem 4 *Let the assumptions of Theorem 3 hold.*

1. *If $\tau_k \to \bar{\tau}$ with*

$$0 < \bar{\tau} < \frac{m}{r_0 L + 1}\min\left\{r_0, \frac{2m}{\kappa_{\mathrm{curv}}^2}\right\}, \tag{22}$$

then x_k converges q-linearly to $\bar{x}$.

2. *If $\tau_k \to 0$, then x_k converges q-superlinearly to $\bar{x}$.*
3. *If $\nabla^2 f(\cdot)$ is Lipschitz continuous on U_0 and $\tau_k \le \tau h_k^{1+\alpha}$ for fixed $\tau > 0$ and $\alpha \ge 0$, then x_k converges q-quadratically to $\bar{x}$.*

Proof The result follows from the proof of [5, Th. 3] with [5, Cor. 2] replaced by Theorem 3. □

Before concluding this section, we provide a technical lemma that is useful for verifying that the subproblem solvers described in the subsequent section satisfy the sufficient decrease conditions (17) or (19).

Lemma 2 *Consider $p : \mathbb{R} \to (-\infty, +\infty]$ defined by $p(t) = \frac{1}{2}\kappa t^2 + \psi(t)$, where $\kappa > 0$ and $\psi : \mathbb{R} \to (-\infty, +\infty]$ is closed, convex and satisfies $\psi(0) = 0$.*

1. *The map $t \mapsto \psi(t)/t$ is nondecreasing on $(0, +\infty)$.*
2. *If there exists $t_0 > 0$ such that $\psi(t_0) < 0$, then there exists $\bar{t} > 0$ such that $p(t) \le \frac{1}{2}\psi(t) < 0$ for all $t \in [0, \bar{t}]$.*
3. *Let $t_\star \in (0, t_1]$ denote a minimizer of p over $[0, t_1]$ for $t_1 > 0$. If there exists $t_0 > 0$ such that $\psi(t_0) < 0$, then $\psi(t_\star) < 0$ and $p(t_\star) \le \frac{1}{2}\psi(t_\star)$.*

Proof To prove the first claim, let $0 < s \le t$ and define $\tau = s/t$. The convexity of ψ and the assumption that $\psi(0) = 0$ ensure that

$$\psi(s)/s = \psi(\tau t)/(\tau t) \le ((1-\tau)\psi(0) + \tau\psi(t))/(\tau t) = \psi(t)/t,$$

as desired.

For the second claim, suppose there exists $t_0 > 0$ such that $\psi(t_0) < 0$. We notice that

$$p(t) \le \tfrac{1}{2}\psi(t) \quad \iff \quad t \le -\psi(t)/(\kappa t).$$

Let $\bar{t} = -\psi(t_0)/(\kappa t_0) > 0$. Then, for any $t \in [0, \bar{t}]$, the first claim ensures that

$$t \le \bar{t} = -\psi(t_0)/(\kappa t_0) \le -\psi(t)/(\kappa t) \quad \implies \quad p(t) \le \tfrac{1}{2}\psi(t) < 0,$$

as desired.

Finally, assume there exists $t_0 > 0$ such that $\psi(t_0) < 0$. The proof of this claim follows, in part, from the optimality of $t_\star$. In particular, we make repeated use of the first-order optimality condition

$$-\kappa t_\star \in \partial(\psi + \iota_{[0,t_1]})(t_\star),$$

which implies

$$\psi(t) \ge \psi(t_\star) + \kappa t_\star(t_\star - t) \qquad \forall\, t \in [0, t_1]. \tag{23}$$

Now, if $t_\star < t_0$, then $\psi(t_\star) < 0$ by the first claim. Otherwise, substituting $t = t_0$ in (23) and noting that $0 < t_0 \le t_\star \le t_1$, we obtain

$$0 > \psi(t_0) \ge \psi(t_\star) + \kappa t_\star(t_\star - t_0) \ge \psi(t_\star).$$

For the second part of this proof, we substitute $t = \frac{1}{2}t_\star$ in (23) to obtain

$$p(t_\star) = \psi(t_\star) + (\kappa t_\star)(t_\star - \tfrac{1}{2}t_\star) \le \psi(\tfrac{1}{2}t_\star) \le \tfrac{1}{2}\psi(t_\star),$$

where the final inequality follows because ψ is convex and satisfies $\psi(0) = 0$. □

Remark 1 (Sufficient Decrease for Iterative Subproblem Solvers) For iterative subproblem solvers, $x_{k,\ell+1}$ typically has the form $x_{k,\ell+1} = x_{k,\ell} + \alpha_{k,\ell}s_{k,\ell}$ for a step $s_{k,\ell} \in X$ and step length $\alpha_{k,\ell} > 0$. In this setting, (19) can be rewritten as

$$p(\alpha_{k,\ell}) \le \mu\psi(\alpha_{k,\ell}),$$

where p is defined in Lemma 2 with κ and ψ given by

$$\kappa = \langle B_k s_{k,\ell}, s_{k,\ell}\rangle \quad \text{and} \quad \psi(t) = t\,\langle \nabla f_k(x_{k,\ell}), s_{k,\ell}\rangle + \phi(x_{k,\ell} + ts_{k,\ell}) - \phi(x_{k,\ell}).$$

Suppose there exists $t_0 > 0$ such that $\psi(t_0) < 0$, then for all $t \in (0, t_0]$, $\psi(t) < 0$ by the first part of Lemma 2. Therefore, if $\kappa \le 0$, then we have that

$$p(t) \le \psi(t) \le \mu\psi(t) \quad \forall\, t \in (0, t_0]$$

and there exists $\alpha_{k,\ell}$ such that (19) holds. On the other hand, if $\kappa > 0$, then Lemma 2 ensures the existence of $\alpha_{k,\ell}$ for which (19) holds. In fact, $\alpha_{k,\ell}$ can be the minimizer of $p(t)$ over some bounded interval $[0, \bar{\alpha}_{k,\ell}]$ for any $\bar{\alpha}_{k,\ell} > 0$.

5 Trust-Region Subproblem Solvers

Motivated by methods for smooth unconstrained and convex-constrained optimization, we introduce three subproblem solvers that generate trial iterates x_k^+ satisfying the FCD condition (8b). The first is a dogleg approach based on [21], the second a simplified version of the algorithm described in [32] that produces trial iterates using the SPG method [9], and the third generalizes the truncated CG method [50, 52]. To achieve guaranteed global convergence as well as rapid local convergence, the trial iterates x_k^+ generated by these methods improve upon a CP. The CP used in [4] is an extension of that used for smooth convex-constrained optimization [32, 37, 53] and satisfies Goldstein-type conditions. To compute this CP, [4, Alg. 2] employs a bidirectional proximal search that typically requires multiple evaluations of the proximity operator of ϕ to satisfy these conditions. To avoid the computational expense of repeatedly evaluating the proximity operator, we introduce a simplified CP based on the SPG step [8] that requires a single evaluation of the proximity operator. This CP is computed as the first iteration of the forthcoming SPG and truncated CG subproblem solvers. On the other hand, the dogleg framework does not depend on a specific CP type, but rather only requires a CP that satisfies (8). In our numerical experiments, the dogleg methods using the CP computed via [4, Alg. 2] tend to outperform those using the simplified CP introduced next.

5.1 Spectral Cauchy Points

We define the simplified CP at the k-th iteration of Algorithm 1 by

$$x_k^c := x_k + \alpha_k(p_k(t_k) - x_k), \tag{24}$$

for $\alpha_k \in [0, 1]$ and $t_k \in [t_{\min}, t_{\max}]$, where $p_k(t)$ is the proximal gradient path

$$p_k(t) := \mathrm{Prox}_{t\phi}(x_k - tg_k) \tag{25}$$

and $0 < t_{\min} \leq t_{\max} < +\infty$ are user-specified parameters. In general, $t_k \in [t_{\min}, t_{\max}]$ can be arbitrary. However, in our numerical examples we choose

$$t_k = \min\{t_{\max}, \max\{t_{\min}, t_{k,0}\}\} \quad \text{for} \quad t_{k,0} := \begin{cases} \frac{\|g_k\|^2}{\langle B_k g_k, g_k\rangle}, \text{ if } \langle B_k g_k, g_k\rangle > 0 \\ \frac{t_0}{\|g_k\|}, \text{ otherwise,} \end{cases}$$

where $t_0 > 0$ is user specified. This specific choice of t_k is related to the SPG or safeguarded Barzilai-Borwein step, where $t_{k,0}$ captures the curvature of B_k in the direction g_k when $\langle B_k g_k, g_k\rangle > 0$. To determine α_k, we first define

$$\alpha_{k,\max} := \min\left\{1, \frac{\Delta_k}{\|s_k\|}\right\},$$

where $s_k := (p_k(t_k) - x_k)$ and then define α_k to be the minimizer of the quadratic upper bound $q_k(\alpha)$, defined by

$$\begin{aligned} m_k(x_k + \alpha s_k) - m_k(x_k) &= f_k(x_k + \alpha s_k) + \phi(x_k + \alpha s_k) - f_k(x_k) - \phi(x_k) \\ &\leq \alpha^2 \tfrac{1}{2} \langle B_k s_k, s_k\rangle + \alpha(\langle g_k, s_k\rangle + \phi(x_k + s_k) - \phi(x_k)) =: q_k(\alpha), \end{aligned} \tag{26}$$

over the interval $[0, \alpha_{k,\max}]$. The upper bound (26) follows from the convexity of ϕ. Note that since $\alpha_k \leq \alpha_{k,\max}$, we have that x_k^c satisfies (8a). In the following proposition, we prove that x_k^c additionally satisfies the FCD condition (8b).

Proposition 2 *Let x_k^c be defined by (24) with arbitrary $t_k \in [t_{\min}, t_{\max}]$ and $\alpha_k \in [0,1]$ given as the minimizer of the quadratic optimization problem*

$$\min_{\alpha\in\mathbb{R}} \; q_k(\alpha) \qquad \text{subject to} \qquad 0 \leq \alpha \leq \alpha_{k,\max}.$$

If $h_k > 0$, then x_k^c satisfies (8) with $\kappa_{\mathrm{fcd}} = \frac{1}{2}\min\{1, t_{\min}\}$ and $\kappa_{\mathrm{rad}} = 1$.

Proof Suppose $h_k > 0$. For simplicity, we define the following quantities

$$\kappa_k := \langle B_k s_k, s_k\rangle \qquad \text{and} \qquad d_k := \langle g_k, s_k\rangle + \phi(x_k + s_k) - \phi(x_k),$$

and note that if $\kappa_k > 0$, then the unconstrained minimizer of q_k is $-d_k/\kappa_k$; recall that $d_k \leq -t_k h_k^2$ from (5). In this case, we have $\alpha_k = \min\{-d_k/\kappa_k, \alpha_{k,\max}\}$. When $\kappa_k = 0$, we have that $q_k(\alpha) = d_k\alpha \leq -t_k h_k^2 \alpha$. Therefore, $\alpha_k = \alpha_{k,\max} > 0$. Finally, if $\kappa_k < 0$, then q_k is concave and α_k is either 0 or $\alpha_{k,\max}$. Considering the two cases that define $\alpha_{k,\max}$, we see that

$$q_k(\alpha_{k,\max}) \leq -h_k \min\{t_k h_k, \Delta_k\} < 0 = q_k(0)$$

and hence $\alpha_k = \alpha_{k,\max}$. This demonstrates that there are three cases we must consider: $\alpha_k = 1$, $\alpha_k = \Delta_k/\|s_k\|$, and $\alpha_k = -d_k/\kappa_k$. The remainder of the proof relies heavily on the bound (26), which in the notation of this proof is

$$m_k(x_k) - m_k(x_k + \alpha s_k) \geq -\tfrac{1}{2}\alpha^2\kappa_k - \alpha d_k = -q_k(\alpha) \quad \forall\, \alpha \in [0,1].$$

Case $\alpha_k = 1$: If $\kappa_k \leq 0$, then

$$m_k(x_k) - m_k(x_k^c) \geq -d_k \geq t_k h_k^2 \geq t_{\min}\frac{h_k^2}{1 + \|B_k\|},$$

where we used the facts that $1 + \|B_k\| \geq 1$ and $t_k \geq t_{\min}$. If $\kappa_k > 0$, then the unconstrained minimizer of q_k, $-d_k/\kappa_k$, is greater than or equal to one and so $-\kappa_k \geq d_k$. Consequently,

$$m_k(x_k) - m_k(x_k^c) \geq -\frac{1}{2}\kappa_k - d_k \geq -\frac{1}{2}d_k \geq \frac{t_{\min}}{2}\frac{h_k^2}{1 + \|B_k\|}.$$

Case $\alpha_k = \Delta_k/\|s_k\|$: If $\kappa_k \leq 0$, then $\alpha_k \leq 1$ and

$$m_k(x_k) - m_k(x_k^c) \geq -\alpha_k d_k \geq \frac{\Delta_k}{\|s_k\|} t_k h_k^2 = \Delta_k h_k.$$

If $\kappa_k > 0$, then $\alpha_k = \Delta_k/\|s_k\| \leq -d_k/\kappa_k$. Consequently,

$$m_k(x_k) - m_k(x_k + \alpha_k s_k) = \alpha_k(-\tfrac{1}{2}\alpha_k\kappa_k - d_k) \geq -\frac{\alpha_k}{2}d_k \geq \frac{1}{2}\Delta_k h_k.$$

Case $\alpha_k = -d_k/\kappa_k$: In this case, $0 < -d_k \leq \kappa_k \leq \|B_k\|\,\|s_k\|^2$ and

$$m_k(x_k) - m_k(x_k^c) \geq \frac{1}{2}\frac{d_k^2}{\kappa_k} \geq \frac{1}{2}\frac{t_k^2 h_k^4}{\|B_k\|\,\|s_k\|^2} \geq \frac{1}{2}\frac{h_k^2}{1 + \|B_k\|}.$$

Combining cases 1, 2 and 3 proves that (8b) holds for x_k^c. □

5.2 Dogleg Subproblem Solver

Dogleg and double dogleg approaches are common trust-region methods that construct piecewise linear paths between the Cauchy and Newton points, and then minimize the quadratic model along these paths. To generalize the dogleg approach to nonsmooth problems of the form (1), we first compute a CP using either the simplified CP from Section 5.1 or [4, Alg. 2], and then compute a Newton point x_k^n that approximately solves trust-region subproblem (2) while ignoring the trust-region constraint:

$$\min_{x\in X} \; f_k(x) + \phi(x). \tag{27}$$

A basic approach to computing the Newton point x_k^n is to apply a finite number of iterations of a descent method to (27), starting at x_k^c. However, if the proximal mapping of ϕ is semismooth, we can instead compute x_k^n by applying a semismooth Newton method [47] to solve the first-order optimality condition

$$x - \mathrm{Prox}_{t\phi}(x - t(B_k(x - x_k) + g_k)) = 0 \tag{28}$$

or the normal mapping equation [49]

$$B_k(\mathrm{Prox}_{t\phi}(z) - x_k) + g_k + t^{-1}(z - \mathrm{Prox}_{t\phi}(z)) = 0 \quad \text{with} \quad x = \mathrm{Prox}_{t\phi}(z). \tag{29}$$

One advantage of (29) over (28) is that $x_k^n \in \mathrm{dom}\,\phi$ by construction. Independent of the approach for generating the Newton point, we assume that x_k^n satisfies the basic model decrease condition

$$m_k(x_k^n) < m_k(x_k^c) < m_k(x_k). \tag{30}$$

We denote the Cauchy and Newton steps by $s_k^c \coloneqq x_k^c - x_k$ and $s_k^n \coloneqq x_k^n - x_k$, respectively. The dogleg algorithm is listed in Algorithm 2.

In our numerical examples, we compute the semismooth Newton step using GMRES preconditioned with a rank-2 perturbation of the identity that approximates the Jacobian. The applications of B_k required by GMRES constitute the main computational expense of the method and could be reduced using problem-specific preconditioners. Beyond this, selecting the simplified CP requires a single evaluation of the proximity operator, while the bidirectional CP may require several evaluations. However, for many problems, evaluating the proximity operator is significantly cheaper than repeatedly solving the semismooth Newton system.

In the following result, we demonstrate that Algorithm 2 produces a viable trial iterate x_k^+ that eventually satisfies Assumption 4, and therefore we can expect rapid convergence via Theorem 4.

Proposition 3 *Algorithm 2 produces a trial iterate x_k^+ that satisfies* (8)*. Moreover, if Assumptions 1, 2 and 3 hold and if we require that the Newton point x_k^n satisfies*

$$m_k(x_k^n) - m_k(x_k) \le \mu\left(\langle g_k, x_k^n - x_k\rangle + \phi(x_k^n) - \phi(x_k)\right), \tag{31a}$$

and

$$H(x_k^n, \nabla f_k(x_k^n), t_k) \le \tau_k h_k, \tag{31b}$$

where μ and τ_k are as in Assumption 4, then Algorithm 1 with subproblem solver Algorithm 2 satisfies Assumption 4 and the convergence rates in Theorem 4 apply.

Algorithm 2 Dogleg Subproblem Solver

Require: The trust-region radius Δ_k and a relaxation parameter $\theta \in (0,1)$ (e.g., $\theta = 0.7$)

1: Compute a generalized Cauchy point x_k^c, and define $s_k^c = x_k^c - x_k$
2: **if** $\|s_k^c\| = \Delta_k$ **then**
3: Return $x_k^+ = x_k^c$
4: **else**
5: Compute a point x_k^n that satisfies (30), and define $s_k^n = x_k^n - x_k$
6: **if** $\|s_k^n\| \leq \Delta_k$ **then**
7: Return $x_k^+ = x_k^n$
8: **else**
9: Compute $\gamma = 1 + \theta(\gamma_0 - 1)$ where $\gamma_0 \in (0,1)$ solves

$$f_k(x_k + \gamma_0 s_k^n) + \gamma_0(\phi(x_k^n) - \phi(x_k)) + \phi(x_k) = m_k(x_k^c)$$

10: **if** $\gamma \|s_k^n\| \leq \Delta_k$ **then**
11: Compute the solution $\alpha_k > 0$ to the quadratic optimization problem

$$\min_{\alpha \in [\gamma, \Delta_k / \|s_k^n\|]} \{f_k(x_k + \alpha s_k^n) + \alpha(\phi(x_k^n) - \phi(x_k)) + \phi(x_k)\}$$

12: Return $x_k^+ = x_k + \alpha_k s_k^n$
13: **else**
14: Compute $\alpha_{k,\max} \in (0,1)$ such that

$$\|s_k^c + \alpha_{k,\max}(\gamma s_k^n - s_k^c)\| = \Delta_k$$

15: Compute the solution $\alpha_k \in [0,1]$ to the quadratic optimization problem

$$\min_{\alpha \in [0, \alpha_{k,\max}]} \{f_k(x_k^c + \alpha(\gamma s_k^n - s_k^c)) + \alpha(\phi(x_k + \gamma s_k^n) - \phi(x_k^c)) + \phi(x_k^c)\}$$

16: Return $x_k^+ = x_k^c + \alpha_k(\gamma s_k^n - s_k^c)$
17: **end if**
18: **end if**
19: **end if**

Proof The root $\gamma_0 \in (0,1)$ in line 9 of Algorithm 2 exists since

$$q(\alpha) = f_k(x_k + \alpha s_k^n) + \alpha(\phi(x_k^n) - \phi(x_k)) + \phi(x_k)$$

is a continuous quadratic polynomial that satisfies

$$q(0) = m_k(x_k) > m_k(x_k^c) > m_k(x_k^n) = q(1).$$

See Figure 1 for an illustration of this fact. Now, if the condition on line 10 holds, then the trial iterate x_k^+ satisfies (8) since

$$m_k(x_k + \alpha s_k^n) \leq q(\alpha) \leq q(\gamma_0) = m_k(x_k^c) \quad \forall\, \alpha \in [\gamma_0, 1],$$

where the first inequality follows from the convexity of ϕ. On the other hand, if the condition on line 10 is violated, then $\alpha_{k,\max}$ in line 14 exists since $\|s_k^c\| < \Delta_k$ and $\gamma \|s_k^n\| > \Delta_k$, and again the convexity of ϕ ensures that

$$m_k(x_k^c + \alpha(\gamma s_k^n - s_k^c)) \leq f_k(x_k^c + \alpha(\gamma s_k^n - s_k^c)) + \alpha(\phi(x_k + \gamma s_k^n) - \phi(x_k^c)) + \phi(x_k^c)$$

for all $\alpha \in [0,1]$. Consequently, (8) holds and Algorithm 2 produces a viable trial iterate.

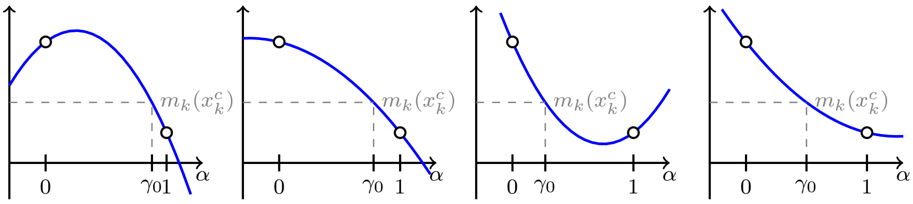


Fig. 1 Possible cases at line 9 of Algorithm 2 when $\langle B_k s_k^n, s_k^n \rangle \neq 0$. The two left images correspond to $\langle B_k s_k^n, s_k^n \rangle < 0$ while the two right images correspond to $\langle B_k s_k^n, s_k^n \rangle > 0$. The blue curve is $q(\alpha)$, which satisfies $q(0) = m_k(x_k)$, $q(\gamma_0) = m_k(x_k^c)$ and $q(1) = m_k(x_k^n)$.

For the second part, since x_k^n satisfies (31), the trial iterate x_k^+ satisfies (17) and (18) from Assumption 4. Therefore, for sufficiently large Δ_k, we have that $x_k^+ = x_k^n$. The result then follows from Theorem 4. □

5.3 Spectral Proximal Gradient Subproblem Solver

Building upon the dogleg approach of Section 5.2, we can improve upon the spectral CP described in Section 5.1 using additional SPG iterations. This approach is closely related to the subproblem solver described in [32] for convex-constrained optimization, which was generalized to our problem class in [4]. In contrast to the subproblem solver described in [4, Alg. 5], our solver does not perform a backtracking linesearch to compute the convex combination parameter α, nor does it require the evaluation of the proximity operator of ϕ augmented with the indicator function of the trust-region constraint. Instead, we compute $\alpha \in [0, 1]$ by minimizing a quadratic upper bound for our model, similar to (26). This subproblem solver is listed in Algorithm 3. The algorithm employs stopping conditions similar to those used in truncated CG for unconstrained problems. In particular, if negative curvature is encountered, the algorithm takes the longest possible step in that direction. Similarly, if the computed step violates the trust-region constraint, then the step is truncated. In addition to these stopping conditions, we terminate Algorithm 3 if the iteration limit `maxit` is exceeded or if the stopping criterion

$$h_{k,\ell} \leq \min\{\bar{\tau}, \tau_k h_{k,0}\} \tag{32}$$

is satisfied for $\bar{\tau} > 0$ and $\tau_k > 0$. Here, $x_{k,\ell}$ is the ℓ-th iterate and

$$h_{k,\ell} := H(x_{k,\ell}, \nabla f_k(x_{k,\ell}), \lambda_{k,\ell}),$$

where $\lambda_{k,\ell} \in [t_{\min}, t_{\max}]$ is the safeguarded spectral step length. During each iteration, Algorithm 3 requires a single evaluation of the proximity operator and a single application of the Hessian. Consequently, Algorithm 3 tends to be more computationally efficient than Algorithm 2.

As in Section 5.2, we now demonstrate that the trial iterates generated by Algorithm 3 eventually satisfy Assumption 4 and therefore recover rapid convergence under Theorem 4.

Proposition 4 *Algorithm 3 produces a trial iterate x_k^+ that satisfies* (8). *Moreover, if Assumptions 1, 2 and 3 hold, then Algorithm 1 with subproblem solver*

Algorithm 3 SPG Trust-Region Subproblem Solver

Require: The initial guess $x_{k,0} = x_k$, $f_{k,0} = f_k(x_k)$, $\phi_{k,0} = \phi(x_k)$, $m_{k,0} = f_{k,0} + \phi_{k,0}$ $d_{k,0} = g_k$, an integer `maxit`, and positive tolerances $\bar{\tau}$ and τ_k, the positive safeguards $t_{\min} \leq t_{\max}$, and $\lambda_{k,0} = t_k \in [t_{\min}, t_{\max}]$

1: Set $\ell = 0$
2: **while** $\ell <$ `maxit` **and** $h_{k,\ell} > \min\{\bar{\tau}, \tau_k h_{k,0}\}$ **and** $\|x_{k,\ell} - x_k\| < \Delta_k$ **do**
3: Set $s \leftarrow \mathrm{Prox}_{\lambda_{k,\ell}\phi}(x_{k,\ell} - \lambda_{k,\ell} d_{k,\ell}) - x_{k,\ell}$
4: Set $\alpha_{\max} \leftarrow 1$
5: **if** $\|x_{k,\ell} + s - x_k\| > \Delta_k$ **then**
6: Set $\alpha_{\max} > 0$ so that $\|x_{k,\ell} + \alpha_{\max} s - x_k\| = \Delta_k$
7: **end if**
8: Compute $\hat{\phi}_{k,\ell} \leftarrow \phi(x_{k,\ell} + s)$, $b \leftarrow B_k s$, and $\kappa \leftarrow \langle b, s\rangle$
9: **if** $\kappa \leq 0$ **then**
10: Set $\alpha \leftarrow \alpha_{\max}$
11: **else**
12: Set $\alpha \leftarrow \min\{\alpha_{\max}, -(\langle d_{k,\ell}, s\rangle + \hat{\phi}_{k,\ell} - \phi_{k,\ell})/\kappa\}$
13: **end if**
14: Set $x_{k,\ell+1} \leftarrow x_{k,\ell} + \alpha s$, $d_{k,\ell+1} \leftarrow d_{k,\ell} + \alpha b$, and $\phi_{k,\ell+1} \leftarrow \phi(x_{k,\ell+1})$
15: **if** $\kappa \leq 0$ **then**
16: Set $\bar{\lambda} \leftarrow t_k / \|d_{k,\ell+1}\|$
17: **else**
18: Set $\bar{\lambda} \leftarrow \langle s, s\rangle / \kappa$
19: **end if**
20: Set $\lambda_{k,\ell+1} \leftarrow \max\{t_{\min}, \min\{t_{\max}, \bar{\lambda}\}\}$
21: Set $\ell \leftarrow \ell + 1$
22: **end while**
23: Return $x_k^+ \leftarrow x_{k,\ell+1}$ as the approximate solution

Algorithm 3 satisfies Assumption 4 and consequently the convergence rates in Theorem 4 apply.

Proof The convexity of ϕ and the definition of s in line 3 of Algorithm 3 ensures that

$$\begin{aligned} m_k(x_{k,\ell} + \alpha s) &= f_k(x_{k,\ell} + \alpha s) + \phi(x_{k,\ell} + \alpha s) \\ &\leq f_k(x_{k,\ell} + \alpha s) + \alpha(\phi(x_{k,\ell} + s) - \phi(x_{k,\ell})) + \phi(x_{k,\ell}). \end{aligned} \tag{33}$$

The upper bound (33) is quadratic in α since f_k is. Consequently, the α computed in lines 4 through 13 in Algorithm 3 is the minimizer of (33) subject to the constraints that $\alpha \in [0, 1]$ and $\|x_{k,\ell} + \alpha s - x_k\| \leq \Delta_k$. One consequence of this is that

$$m_k(x_{k,\ell+1}) \leq m_k(x_{k,\ell}) \qquad \forall\, \ell = 1, 2, \ldots.$$

Since the first step $x_{k,1} = x_k^c$, where x_k^c is the Cauchy point defined in (24), we have that (8) is satisfied.

For the second part, we proceed similarly to the proof of Proposition 2. We demonstrate that (19) is satisfied by considering three cases: $\alpha = 1$, α solves $\|x_{k,\ell} + \alpha s - x_k\| = \Delta_k$ and $\alpha = -(\langle d_{k,\ell}, s\rangle + \hat{\phi}_{k,\ell} - \phi_{k,\ell})/\kappa = -\psi(1)/\kappa$, where ψ and κ are specified in Remark 1. As described in Remark 1, if $\kappa \leq 0$ then (19) holds since the SPG step satisfies $\psi(1) < 0$. Now suppose $\kappa > 0$. If $\alpha = 1$, then $1 < -\psi(1)/\kappa$ or equivalently $\kappa < -\psi(1)$, which ensures that

$$p(1) = \tfrac{1}{2}\kappa + \psi(1) < \tfrac{1}{2}(-\psi(1)) + \psi(1) = \tfrac{1}{2}\psi(1).$$

In the second case, we have that $\alpha \leq \min\{1, -\psi(1)/\kappa\}$ and so $\kappa\alpha \leq -\psi(1)$ and $-\psi(\alpha)/\alpha \geq -\psi(1)$ by Lemma 2. These two facts imply

$$p(\alpha) = \tfrac{1}{2}\kappa\alpha^2 + \psi(\alpha) \leq \tfrac{1}{2}\alpha(-\psi(1)) + \psi(\alpha) \leq \tfrac{1}{2}\alpha(-\psi(\alpha)/\alpha) + \psi(\alpha) = \tfrac{1}{2}\psi(\alpha).$$

Finally, if $\alpha = -\psi(1)/\kappa$, then $-\psi(1)/\kappa \leq 1$ and

$$p(\alpha) = \tfrac{1}{2}\psi(1)^2/\kappa + \psi(\alpha) \leq \tfrac{1}{2}\alpha(-\psi(1)) + \psi(\alpha) \leq \tfrac{1}{2}\alpha(-\psi(\alpha)/\alpha) + \psi(\alpha) = \tfrac{1}{2}\psi(\alpha).$$

Consequently, (19) is satisfied and we can expect rapid convergence from Theorem 4. □

5.4 Nonlinear Conjugate Gradient Subproblem Solver

Motivated by its efficiency for solving smooth unconstrained [50,52] and constrained [30,31,37] optimization problems, we extend the truncated CG algorithm to solve (2) with the potentially nonsmooth nonquadratic term ϕ. There are three locations in the truncated CG algorithm that must be modified: first, we replace the negative gradient computed at each iteration with the SPG step

$$p_{k,\ell} = \frac{1}{\lambda_{k,\ell}}(\mathrm{Prox}_{\lambda_{k,\ell}\phi}(x_{k,\ell} - \lambda_{k,\ell}\nabla f_k(x_{k,\ell})) - x_{k,\ell}), \tag{34}$$

where $x_{k,\ell}$ denotes the ℓ-th CG iteration and $\lambda_{k,\ell} \in [t_{\min}, t_{\max}]$ is the safeguarded spectral step length (see lines 15-19 in Algorithm 3); second, we replace the exact line search with an iterative one since the model m_k is not necessarily quadratic; and third, we select the conjugacy parameter β using a nonlinear CG rule such as the nonnegative Dai-Yuan parameter [20]

$$\beta_{k,\ell} = \max\left\{0, \frac{\|p_{k,\ell}\|^2}{\langle p_{k,\ell-1} - p_{k,\ell}, s_{k,\ell-1}\rangle}\right\}.$$

Other such updates can be used, e.g. Fletcher-Reeves, Polak-Ribière, Hestenes-Stiefel, etc. The cost of these modifications is modest; as in Section 5.3, we only require a single proximity operator evaluation and a single application of the Hessian per CG iteration.

During the line search, we incur the additional cost of repeatedly evaluating the nonsmooth term ϕ. This procedure determines the step length $\alpha > 0$ that approximately minimizes the one-dimensional function

$$q_{k,\ell}(\alpha) := m_k(x_{k,\ell} + \alpha s_{k,\ell}).$$

To determine α, we first minimize the quadratic upper bound of $q_{k,\ell}$

$$q_{k,\ell}(t\gamma_{k,\ell}) \leq f_k(x_{k,\ell} + t\gamma_{k,\ell}s_{k,\ell}) + t(\phi(x_{k,\ell} + \gamma_{k,\ell}s_{k,\ell}) - \phi(x_{k,\ell})) + \phi(x_k, \ell), \tag{35}$$

for $t \in [0,1]$, where $\gamma_{k,\ell} \in (0, \bar{\alpha}_{k,\ell}]$ is chosen so that $\phi(x_{k,\ell} + \gamma_{k,\ell}s_{k,\ell}) < +\infty$ and $\bar{\alpha}_{k,\ell} > 0$ is chosen so that

$$\|x_{k,\ell} + \bar{\alpha}_{k,\ell}s_{k,\ell} - x_k\| = \Delta_k. \tag{36}$$

The upper bound in (35) follows from the convexity of ϕ. Since (35) is quadratic, we can compute the exact minimizer, $t_{k,\ell}$. Using this minimizer, we define the initial guess $\alpha_{k,\ell}^0 := t_{k,\ell}\gamma_{k,\ell}$. We then approximately minimize $q_{k,\ell}$ using finitely many iterations of Brent's method [11], which produces the step length $\alpha_{k,\ell}$. Since Brent's method yields a sequence of decreasing function values, we have that $q_{k,\ell}(\alpha_{k,\ell}) \le q_{k,\ell}(\alpha_{k,\ell}^0)$. In addition, we terminate Brent's method when the computed step satisfies (19), i.e., $\alpha_{k,\ell}$ satisfies

$$q_{k,\ell}(\alpha_{k,\ell}) - q_{k,\ell}(0) \le \mu(\alpha_{k,\ell}\langle \nabla f_k(x_{k,\ell}), s_{k,\ell}\rangle + \phi(x_{k,\ell} + \alpha_{k,\ell}s_{k,\ell}) - \phi(x_{k,\ell})). \quad (37)$$

Similar to other nonlinear CG methods, we employ restarts. That is, we set $\beta_{k,\ell} = 0$ (i.e., revert to the SPG step) if the current step $s_{k,\ell}$ does not produce sufficient decrease [46] defined by the inequality

$$\langle \nabla f_k(x_{k,\ell}), s_{k,\ell}\rangle + \phi(x_{k,\ell} + s_{k,\ell}) - \phi(x_{k,\ell}) > -(1-\eta)\,\|p_{k,\ell}\|^2 . \quad (38)$$

With (38) in mind, we set

$$\gamma_{k,\ell} := \begin{cases} \min\{\bar{\alpha}_{k,\ell}, \lambda_{k,\ell}\} & \text{if } \ell = 0 \text{ or } \beta_{k,\ell} = 0, \\ \min\{\bar{\alpha}_{k,\ell}, 1\} & \text{otherwise,} \end{cases} \quad (39)$$

which ensures that $\phi(x_{k,\ell} + \alpha_{k,\ell}^0 s_{k,\ell}) < +\infty$. We list the complete routine in Algorithm 4. In the following proposition, we demonstrate that the trial iterates

Algorithm 4 Truncated Nonlinear CG Trust-Region Subproblem Solver

Require: The initial guess $x_{k,0} = x_k$, $d_{k,0} = g_k$, an integer `maxit`, and positive tolerances $\bar{\tau}$, τ_k, $t_{\min} \le t_{\max}$, $\lambda_{k,0} = t_k \in [t_{\min}, t_{\max}]$, and $\eta \in (0, 1]$

Set $\ell \leftarrow 0$
Set $p_{k,0} \leftarrow (\mathrm{Prox}_{\lambda_{k,0}\phi}(x_{k,0} - \lambda_{k,0}d_{k,0}) - x_{k,0})/\lambda_{k,0}$ and $s_{k,0} \leftarrow p_{k,0}$
Compute $h_{k,0} \leftarrow \|p_{k,0}\|$
while $\ell <$ `maxit` **and** $h_{k,\ell} > \min\{\bar{\tau}, \tau_k h_{k,0}\}$ **and** $\|x_{k,\ell} - x_k\| < \Delta_k$ **do**
  Set $b_{k,\ell} \leftarrow B_k s_{k,\ell}$ and $\kappa_{k,\ell} \leftarrow \langle b_{k,\ell}, s_{k,\ell}\rangle$
  Compute $\alpha_{k,\ell} \in (0, \bar{\alpha}_{k,\ell}]$ that satisfies (37), where $\bar{\alpha}_{k,\ell}$ is the positive root of (36)
  Set $x_{k,\ell+1} \leftarrow x_{k,\ell} + \alpha_{k,\ell}s_{k,\ell}$ and $d_{k,\ell+1} \leftarrow d_{k,\ell} + \alpha_{k,\ell}b_{k,\ell}$
  if $\kappa_{k,\ell} \le 0$ **then**
    Set $\bar{\lambda} \leftarrow t_k / \|d_{k,\ell+1}\|$
  else
    Set $\bar{\lambda} \leftarrow \|s_{k,\ell}\|^2 / \kappa_{k,\ell}$
  end if
  Set $\ell \leftarrow \ell + 1$
  Set $\lambda_{k,\ell} \leftarrow \max\{t_{\min}, \min\{t_{\max}, \bar{\lambda}\}\}$
  Set $p_{k,\ell} \leftarrow (\mathrm{Prox}_{\lambda_{k,\ell}\phi}(x_{k,\ell} - \lambda_{k,\ell}d_{k,\ell}) - x_{k,\ell})/\lambda_{k,\ell}$
  Set $h_{k,\ell} \leftarrow \|p_{k,\ell}\|$
  Set $\beta_{k,\ell} \leftarrow \max\{0, h_{k,\ell}^2 / \langle p_{k,\ell-1} - p_{k,\ell}, s_{k,\ell-1}\rangle\}$
  Set $s_{k,\ell} \leftarrow p_{k,\ell} + \beta_{k,\ell}s_{k,\ell-1}$
  if $\langle d_{k,\ell}, s_{k,\ell}\rangle + \phi(x_{k,\ell} + s_{k,\ell}) - \phi(x_{k,\ell}) > -(1-\eta)\,\|p_{k,\ell}\|^2$ **then**
    Set $s_{k,\ell} \leftarrow p_{k,\ell}$ and $\beta_{k,\ell} \leftarrow 0$
  end if
end while
Return $x_k^+ \leftarrow x_{k,\ell}$ as the approximate solution

generated by Algorithm 4 are viable and, as before, yield rapid convergence under Theorem 4. The latter result essentially follows from the restart procedure.

Proposition 5 *Algorithm 4 produces a trial iterate x_k^+ that satisfies* (8). *Moreover, if Assumptions 1, 2 and 3 hold, then Algorithm 1 with subproblem solver Algorithm 4 satisfies Assumption 4 and consequently the convergence rates in Theorem 4 hold.*

Proof Since the step length defined in Proposition 2 is feasible with respect to the one-dimensional minimization problem considered here, we have that each step of Algorithm 4 satisfies (8).

For the second part, reverting to the SPG step when (38) is satisfied ensures that

$$\psi(t) = t\,\langle \nabla f_k(x_{k,\ell}), s_{k,\ell}\rangle + \phi(x_{k,\ell} + t s_{k,\ell}) - \phi(x_{k,\ell}) < 0$$

with $t = \lambda_{k,\ell}$ if (38) holds and with $t = 1$ otherwise. As a consequence, Lemma 2 ensures that there exists an $\alpha_{k,\ell}$ such that (37) is satisfied. Hence, (17) is satisfied by Algorithm 4 and the result follows from Theorem 4. □

6 Numerical Results

We now apply the aforementioned methods to an array of applications. While we hesitate to name a *best* solver for all problems, we compare them against [4, Alg. 5] and by proxy other nonsmooth methods including FISTA, nmAPG, PANOC, etc., cf. [4, Sect. 5]. We demonstrate the performance of each subproblem solver on six numerical examples: the first three arising from data science and the final three from PDE-constrained optimization. For future reference, the support vector machine, semilinear optimal control and topology optimization examples were also used as tests in [4, Sect. 5].

Low-Rank Matrix Completion (`LowRank`). Our first example is the rank minimization problem

$$\min_{X\in\mathbb{R}^{M\times N}} \; \frac{1}{2}\,\|\mathcal{A}X - Y\|_F^2 + \|X\|_*\,, \tag{40}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_*$ is the nuclear norm, and $\mathcal{A}$ is a selection matrix that observes 50% of the matrix entries [2,14,48]. In our example, $M = N = 225$ and Y is the observed data, which we corrupt with additive Gaussian noise (mean zero and variance 0.01). The matrix used to generate Y has rank 25. Recall that the nuclear norm has a computable proximity operator that is semismooth [22].

Support Vector Machine (`SVM`). Our second example is the nonconvex support vector machine problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m}\{1 - \tanh(b_i\langle a_i, x\rangle)\} + \lambda\|x\|_1, \tag{41}$$

where $\lambda > 0$, $b_i \in \{-1, 1\}$ are labels, and $a_i \in \mathbb{R}^n$ are data points for $i = 1, \ldots, m$. This problem was studied in [17] and is a nonsmooth extension of the problems considered in [40,54]. We use the `phishing` data set [24] from the `LIBSVM` data repository [15]. The number of data points is $m = 11{,}055$ and the number of features is $n = 68$. We set the regularization parameter to $\lambda = 10^{-2}$.

Logistic Regression (`Logistic`). Our third example is the sparse logistic regression problem

$$\min_{x_w\in\mathbb{R}^n,\, x_v\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m}\{\log(1+\exp(-b_i(\langle a_i, x_w\rangle + x_v)))\} + \lambda\|x\|_1, \tag{42}$$

where $\lambda > 0$, $b_i \in \{-1, 1\}$ are labels, and $a_i \in \mathbb{R}^n$ are data points for $i = 1, \ldots, m$. This problem has been used for classification [26,38,25]. We solve (42) for various datasets from the `LIBSVM` data repository.

Optimal Control of Burgers' Equation (`Burgers`). Our fourth example is the optimal control of Burgers' equation

$$\min_{z\in L^2(\Omega)} \frac{1}{2}\int_\Omega ([S(z)](x) - w(x))^2 \mathrm{d}x + \frac{\alpha}{2}\int_\Omega z(x)^2\,\mathrm{d}x + \beta\int_\Omega |z(x)|\,\mathrm{d}x \tag{43a}$$

where $\Omega = (0, 1)$ is the physical domain, $\alpha = 10^{-4}$ and $\beta = 10^{-2}$ are penalty parameters, $w(x) = -x^2$ is the target state, and $S(z) = u \in H^1(\Omega)$ solves the weak form of Burgers' equation

$$\begin{aligned} -\nu u'' + u\,u' &= z + f \quad \text{in } \Omega, \\ u(0) = 0, \quad u(1) &= -1, \end{aligned} \tag{44}$$

where $f(x) = 2(\nu + x^3)$ and $\nu = 0.08$. We discretize the state u using continuous piecewise linear finite elements and the control z using piecewise constants on a uniform mesh with $n = 512$ intervals.

Semilinear Optimal Control (`Semilinear`). Our fifth example is the optimal control of a semilinear elliptic PDE

$$\min_{z\in L^2(\Omega)} \frac{1}{2}\int_\Omega ([S(z)](x) - w(x))^2 \mathrm{d}x + \frac{\alpha}{2}\int_\Omega z(x)^2\,\mathrm{d}x + \beta\int_\Omega |z(x)|\,\mathrm{d}x \tag{45a}$$

$$\text{subject to} \qquad -25 \le z \le 25 \ \text{ a.e.}, \tag{45b}$$

where $\Omega = (0, 1)^2$ is the physical domain, $\alpha = 10^{-4}$ and $\beta = 10^{-2}$ are penalty parameters, $w \equiv -1$ is the target state, and $u = S(z) \in H^1(\Omega)$ solves the weak form of the semilinear elliptic PDE

$$-\Delta u + u^3 = z \quad \text{in } \Omega \tag{46a}$$

$$u = 0 \quad \text{on } \partial\Omega. \tag{46b}$$

We discretize the state u using continuous piecewise linear finite elements on a uniform triangular mesh with 131,072 elements and the control variable z using piecewise constants on the same mesh, resulting in 131,072 degrees of freedom.

Topology Optimization (`TopOpt`). Our final example is the compliance minimization problem

$$\min_{\rho\in L^2(\Omega)} \int_{\Gamma_t} T(x)[S(\rho)](x)\,\mathrm{d}x \tag{47a}$$

$$\text{subject to} \quad \int_{\Omega} \rho(x)\,\mathrm{d}x = v|\Omega|, \quad 0 \le \rho \le 1 \ \text{ a.e.}, \tag{47b}$$

where $\Omega = (0,150)\times(0,50)$ is the physical domain, $v = 0.4$ is the volume fraction, $\Gamma_d = \{0\}\times[0,50]$ is the fixed boundary, $\Gamma_t = \partial\Omega\setminus\Gamma_d$ is the traction boundary and $S(\rho) = u \in H^1(\Omega)^2$ solves the weak form of the linear elasticity equations

$$-\nabla\cdot(K(\rho):\varepsilon) = 0 \qquad \text{in } \Omega \tag{48a}$$

$$\varepsilon = \tfrac{1}{2}(\nabla u + \nabla u^\top) \qquad \text{in } \Omega \tag{48b}$$

$$K(\rho):\varepsilon\mathbf{n} = T \qquad \text{in } \Gamma_t \tag{48c}$$

$$u = 0 \qquad \text{in } \Gamma_d. \tag{48d}$$

Here, $\mathbf{n}$ denotes the outward pointing normal vector,

$$K(\rho) \coloneqq [\kappa_{\min} + (1-\kappa_{\min})\mathbb{F}(\rho)^3]K_0,$$

K_0 is the usual isotropic elasticity matrix, $\mathbb{F}$ is the Helmholtz filter [36] with filter radius 0.1, $\kappa_{\min} = 10^{-4}$, and $T \in L^2(\Gamma_t)^2$ is the traction force: $T(x) = (0,0)$ for $x \in \Gamma_t\setminus(\{150\}\times[0,1])$ and $T(x) = (0,-1)$ for $x \in \{150\}\times[0,1]$. For our numerical results, the Young's modulus is 200 and the Poisson ratio is 0.29. We discretize the displacements u and the filtered density $\mathbb{F}(\rho)$ using continuous piecewise linear finite elements on a 150×50 uniform quadrilateral mesh, and the density ρ using piecewise constants on the same mesh, resulting in 7,500 degress of freedom.

For each example, we employ the quadratic model (7) with $g_k = \nabla f(x_k)$ and $B_k = \nabla^2 f(x_k)$, making Algorithm 1 an inexact proximal Newton method that rigorously handles indefinite Hessians. We test up to six subproblem solvers, depending on the nonsmooth term ϕ:

- `SPG` is the spectral proximal gradient solver described in [4, Alg. 5];
- `SPG2` is the spectral proximal gradient solver Algorithm 3;
- `NCG` is the nonlinear CG solver Algorithm 4;
- `SEMI` is Algorithm 2 with the Newton point computed via (28)[1];
- `NORM` is Algorithm 2 with the Newton point computed via (29)[1];
- `OBM` is the L^1-specific solver described in Appendix A.

As demonstrated in [4, Sect. 5], `SPG` outperformed various competing methods, reducing the time-to-solution by factors between 7x and 70x. We use the following algorithmic parameters for all examples: $\Delta_0 = 50$, $\eta_1 = 0.05$, $\eta_2 = 0.9$, $\gamma_1 = \gamma_2 = 0.25$, $\gamma_3 = 2.5$, $\mu_1 = 10^{-4}$, $\beta_{\mathrm{dec}} = 0.1$, and $\beta_{\mathrm{inc}} = 10$. We employ the bidirectional CP algorithm [4, Alg. 2] for `SEMI`, `NORM` and `OBM`, and allow at most two iterations of increase. This choice of CP resulted in the best performance for these subproblem solvers when compared with the simplified CP (24). We stop Algorithm 1 if $h_k \le 10^{-5}$ and we stop the iterative subproblem solvers using the condition (32) with the absolute tolerance $\bar{\tau} = 10^{-5}$ and tolerance sequence

[1] This method requires that the proximity operator is semismooth.

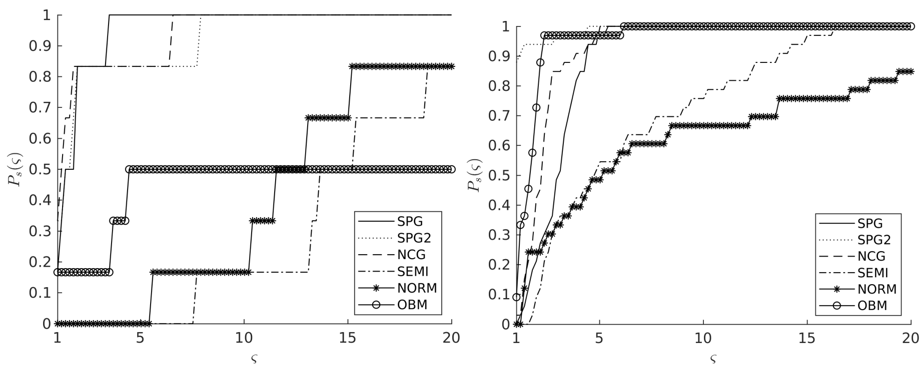
Example	AlgType	iter	fval	grad	hess	phi	prox	time (s)
RankMin	SPG	4	5	5	40	99	111	1.80
	SPG2	4	5	5	49	67	52	0.91
	NCG	8	9	7	89	2053	179	3.67
	SEMI	3	4	4	483	25	178	17.25
	NORM	2	3	3	475	17	173	15.77
SVM	SPG	21	22	18	231	603	695	0.64
	SPG2	31	32	28	406	591	425	0.97
	NCG	22	23	16	162	3251	310	0.46
	SEMI	20	21	20	3668	135	1716	7.53
	NORM	12	13	12	2364	79	1160	4.78
	OBM	78	79	74	607	1107	440	1.49
Logistic	SPG	9	10	10	153	401	208	0.45
	SPG2	18	19	19	306	457	325	0.52
	NCG	10	11	9	123	2513	247	0.37
	SEMI	23	24	23	4385	2315	151	4.15
	NORM	16	17	17	3907	107	2625	4.12
	OBM	70	71	70	568	1043	396	1.04
Burgers	SPG	11	12	8	127	331	3940	0.28
	SPG2	13	14	10	154	208	158	0.11
	NCG	15	16	10	115	2033	220	0.13
	SEMI	18	19	15	3828	173	1863	1.24
	NORM	12	13	9	2773	99	881	0.76
	OBM	14	15	11	101	117	81	0.08
Semilinear	SPG	2	3	3	34	91	67	8.67
	SPG2	2	3	3	34	51	37	8.49
	NCG	2	3	3	32	1469	67	9.39
	SEMI	8	9	9	1331	59	333	68.84
	NORM	6	7	7	978	45	299	52.53
TopOpt	SPG	15	16	15	201	463	509	5.74
	SPG2	106	107	105	1735	1860	1835	43.92
	NCG	20	21	19	269	11987	543	7.89
	SEMI	86	87	86	18052	577	20644	401.88
	NORM	109	110	105	34309	739	26213	712.46

Table 1 Algorithmic performance for all examples: `iter` is the number of trust-region iterations, `fval` and `grad` are the numbers of f and ∇f evaluations, respectively, `hess` is the number of $\nabla^2 f$ applications, `phi` is the number of ϕ evaluations, `prox` is the number of proximity operator evaluations, and `time (s)` is the total wallclock time in seconds.

$\tau_k = 10^{-3} h_k$. We set the maximum number of iterations for each subproblem solver to 15. For `NCG`, we set the number of Brent's iterations in (37) to 10. For `SEMI` and `NORM`, we solve (28) and (29), respectively, using semismooth Newton, globalized with a line search. We compute the semismooth Newton step using GMRES with a maximum of 10 iterations and precondition the solve with a rank-2 perturbation of the identity similar to BFGS. For `OBM`, we set $\texttt{maxit} = 1$ and $\texttt{maxitcg} = 5$.

We summarize the performance of all subproblem algorithms in Table 1, where we tabulate the number of trust-region iterations (`iter`), the number of f (`fval`) and ∇f (`grad`) evaluations, the number of $\nabla^2 f$ applications (`hess`), the number of ϕ evaluations (`phi`), the number of proximity operator evaluations (`prox`), and the wallclock time in seconds (`time (s)`). We additionally include performance profiles in the style of [23]. We use the performance ratio $\varrho_{p,s}$ given by

$$\varrho_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s'} \mid s' \in \mathcal{S}\}},$$



(a) Performance plot for the Table 1 examples. (b) Performance plot for `Logistic` using various `LIBSVM` datasets.

Fig. 2 Algorithmic performance for all examples with $\varsigma \in [1, 20]$. For the examples in Table 1 (Figure 2(a)), the win rates are $P_s(1) = \{1/6, 1/3, 1/3, 0, 0, 1/6\}$ corresponding to the solvers $\mathcal{S} = \{$`SPG, SGP2, NCG, SEMI, NORM, OBM`$\}$. For the `Logistic` example (Figure 2(b)), the win rates are $P_s(1) = \{0.0303, 0.8788, 0, 0, 0, 0.0909\}$.

where $t_{p,s}$ is the wallclock time in seconds required to solve problem p using solver s. We denote the set of solvers by $\mathcal{S} := \{1, \ldots, n_s\}$ and a set of problems by $\mathcal{P} := \{1, \ldots, n_p\}$. The performance profile represents the probability that a given solver's performance ratio is within a factor $\varsigma \in \mathbb{R}$ of the best possible ratio, i.e.,

$$P_s(\varsigma) := \frac{1}{n_p} \text{size} \{p \in \mathcal{P} \mid \varrho_{p,s} \leq \varsigma\} .$$

The performance ratio of a solver $s \in \mathcal{S}$ that cannot solve problem p is given the (arbitrary) value $\varrho_{p,s} = \varrho_M = 1000$ (e.g., since `OBM` is an L^1-specific solver, its performance ratio is assigned ϱ_M for `RankMin`, `Semilinear` and `TopOpt`). If the set of problems $\mathcal{P}$ is large, then solvers with large P_s are preferred. When $\mathcal{P}$ is small, one can draw conclusions for that set of problems by comparing the win percentage $P_s(1)$. Figure 2 encompasses two plots: Figure 2(a) ($n_s = 6$, $n_p = 6$) displays the performance profile for all the solvers on the examples summarized in Table 1 with the `Logistic` example using the `phishing` data set; and Figure 2(b) ($n_s = 6$, $n_p = 33$) depicts the solver performance on the `Logistic` example only, but using all classification datasets from the `LIBSVM` data repository [16] that were not overly large—we excluded any data set that was stored in a compressed format like `.zip`, `.bz`, `.xz`, or `.tar`. The 33 datasets that we used are listed in the data availability statement.

In Table 1, we observe that all subproblem solvers perform comparably in terms of total trust-region iterations on all problems with the exception of `TopOpt`. As one might expect, the dogleg methods `SEMI` and `NORM` require the most Hessian applications, which are used to iteratively compute the Newton points. In many of the examples, the application of the Hessian is the dominant cost, which is reflected in the computational times for `SEMI` and `NORM` as depicted in Figure 2(a). On the other hand, `NCG` requires the most evaluations of ϕ, which is required for the Brent's line search. The cost of these evaluations is especially apparent on `RankMin`, where evaluating ϕ requires the computation of a singular value decomposition. As a general trend, `SPG`, `SPG2` and `NCG` tend to outperform the other methods with respect to wallclock time because they require fewer applications of the Hessian and proximity operator; this is reiterated in Figure 2. It is worth noting that SPG2 has the lowest cost per iteration. However, it is not competitive on TopOpt. A possible reason is that the smooth objective function in TopOpt is highly nonconvex, resulting in a model Hessian B_k with many directions of negative curvature. Consequently, the spectral step size is rarely used when selecting $\lambda_{k,\ell}$. In contrast, SPG works quite well on TopOpt. The main difference between SPG and SPG2 is the CP, suggesting that the CP computed via bidirectional proximal search [4, Alg. 5] produces a better step for TopOpt then the simplified CP defined by (24). In Figure 2(b), we see that SPG2 attains the highest win percentage $P_s(1)$ and is a close second in achieving $P_s(\varsigma) = 1$. The other SPG-based methods perform comparably on all data sets for Logistic and the L^1-specific algorithm OBM performs notably well. Again, the dogleg methods clearly perform worse than the other methods since they are hindered by the expensive Hessian evaluations for solving the semismooth Newton system. Of course, this cost can be reduced with good preconditioners for the semismooth Newton system. However, such preconditioners would be problem dependent and are beyond the scope of this paper.

7 Conclusion

We have introduced three new subproblem solvers for the trust-region algorithm introduced in [4, Alg. 1], each generalizing a smooth counterpart, that are generally comparable to the SPG-based solver described in [4]. Our methods provide guaranteed rapid local convergence under specific assumptions on the problem data. Moreover, we have demonstrated the performance of these subproblem solvers on six numerical examples taken from data science and PDE-constrained optimization. While there is no clearly superior method, we do generally see that the broader class of methods based on SPG (i.e., SPG, SPG2, and NCG) tend to excel for the applications studied. Although our numerical results are inconclusive regarding a clear winner, we expect that each method may have a niche application on which it outperforms the others. In addition, we expect that the performance of the individual methods can be improved by modifying or replacing certain expensive components like Brent's method in NCG or GMRES in SEMI and NORM. We leave this as future research.

Statements and Declarations

Competing Interests: The authors declare that they have no conflicts of interest.
Data Availability: The data sets used for the SVM and Logistic examples were downloaded from the LIBSVM data repository and include: a1a-a9a, australian, breast-cancer, diabetes, fourclass, german.numer, heart, ionosphere_scale, liver-disorders, madelon, mushrooms, phishing, skin_nonskin, sonar_scale, splice, svmguide1, svmguide3, and w1a-w8a. The other data generated for the numerical results are available from the corresponding author upon reasonable request.

References

1. Andrew, G., Gao, J.: Scalable training of L_1-regularized log-linear models. In: 24th International Conference on Machine Learning, pp. 33–40. ACM (2007)
2. Aravkin, A.Y., Burke, J.V., Drusvyatskiy, D., Friedlander, M.P., Roy, S.: Level-set methods for convex optimization. Mathematical Programming Series B **174**, 359–390 (2019)
3. Babuška, I.: Error-bounds for finite element method. Numerische Mathematik **16**(4), 322–333 (1971)
4. Baraldi, R.J., Kouri, D.P.: A proximal trust-region method for nonsmooth optimization with inexact function and gradient evaluations. Mathematical Programming **201**(1), 1–40 (2022)
5. Baraldi, R.J., Kouri, D.P.: Local convergence analysis of an inexact trust-region method for nonsmooth optimization. Optimization Letters p. submitted (2023)
6. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. CMS Books in Mathematics. Springer International Publishing, Cham, Switzerland (2018)
7. Bellavia, S., Gurioli, G., Morini, B., Toint, P.: The impact of noise on evaluation complexity: The deterministic trust-region case. Journal of Optimizaton Theory and Applications **196**, 700–729 (2023)
8. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10**(4), 1196–1211 (2000)
9. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. J. Stat. Softw **60**(3), 1–21 (2014)
10. Bolte, J., Daniilidis, A., Lewis, A.: Tame functions are semismooth. Mathematical Programming **117**, 5–19 (2009)
11. Brent, R.P.: Algorithms for minimization without derivatives. Courier Corporation (2013)
12. Brezzi, F.: On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. Publications des séminaires de mathématiques et informatique de Rennes (S4), 1–26 (1974)
13. Byrd, R.H., Nocedal, J., Oztoprak, F.: An inexact successive quadratic approximation method for L-1 regularized optimization. Mathematical Programming **157**(2), 375–396 (2016)
14. Candés, E.J., Recht, B.: Exact matrix completion via convex optimization. Foundations of Computational Mathematics **9**, 717–772 (2009)
15. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(3), 1–27 (2011). Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
16. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011). Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
17. Chen, Z., Milzarek, A., Wen, Z.: A trust-region method for nonsmooth nonconvex optimization (2021)
18. Conn, A.R., Gould, N.I.M., Sartenaer, A., Toint, P.L.: Convergence properties of minimization algorithms for convex constraints using a structured trust region. SIAM Journal on Optimization **6**(4), 1059–1086 (1996). DOI 10.1137/S1052623492236481
19. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust region methods. SIAM, Philadelphia, PA (2000)
20. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on optimization **10**(1), 177–182 (1999)
21. Dennis Jr., J.E., Mei, H.H.W.: Two new unconstrained optimization algorithms which use function and gradient values. J. Optim. Theory and Applics. **28**, 453–482 (1979). DOI 10.1007/BF00932218
22. Ding, C., Sun, D., Sun, J., Toh, K.C.: Spectral operators of matrices: Semismoothness and characterizations of the generalized jacobian. SIAM Journal of Optimization **30**(1), 630–659 (2020)
23. Dolan, E.D., Moré, J.: Benchmarking optimization software with performance profiles. Mathematical Programming Series A **91**, 201–213 (2002)
24. Dua, D., Graff, C.: UCI machine learning repository (2017). URL http://archive.ics.uci.edu/ml
25. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l 1-ball for learning in high dimensions. In: Proceedings of the 25th international conference on Machine learning, pp. 272–279 (2008)

26. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression (2004)
27. Garreis, S., Ulbrich, M.: An inexact trust-region algorithm for constrained problems in Hilbert space and its application to the adaptive solution of optimal control problems with PDEs. Preprint, submitted, Technical University of Munich (2019)
28. Golub, G.H., von Matt, U.: Quadratically constrained least squares and quadratic problems. Numerische Mathematik **59**, 561–580 (1991)
29. Gould, N.I.M., Lucidi, S., Roma, M., Toint, P.L.: Solving the trust-region subproblem using the lanczos method. SIAM Journal on Optimization **9**(2), 504–525 (1999). DOI 10.1137/S1052623497322735
30. Heinkenschloss, M., Ridzal, D.: A matrix-free trust-region SQP method for equality constrained optimization. SIAM Journal on Optimization **24**(3), 1507–1541 (2014)
31. Kelley, T., Sachs, E.: A trust region method for parabolic boundary control problems. SIAM Journal on Optimization **9**(4), 1064–1081 (1999)
32. Kouri, D.P.: A matrix-free trust-region Newton algorithm for convex-constrained optimization. Optimization Letters pp. 1–15 (2021)
33. Kouri, D.P., Heinkenschloss, M., Ridzal, D., van Bloemen Waanders, B.G.: A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. SIAM Journal on Scientific Computing **35**(4), A1847–A1879 (2013)
34. Kouri, D.P., Heinkenschloss, M., Ridzal, D., van Bloemen Waanders, B.G.: Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty. SIAM Journal on Scientific Computing **36**(6), A3011–A3029 (2014)
35. Kouri, D.P., Ridzal, D.: Inexact trust-region methods for PDE-constrained optimization. In: Frontiers in PDE-Constrained Optimization, pp. 83–121. Springer, New York, NY (2018)
36. Lazarov, B.S., Sigmund, O.: Filters in topology optimization based on Helmholtz-type differential equations. International Journal for Numerical Methods in Engineering **86**(6), 765–781 (2011)
37. Lin, C.J., Moré, J.J.: Newton's method for large bound-constrained optimization problems. SIAM Journal on Optimization **9**(4), 1100–1127 (1999)
38. Liu, J., Chen, J., Ye, J.: Large-scale sparse logistic regression. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 547–556 (2009)
39. Maciel, M.C., Mendonça, M.G., Verdiell, A.B.: Monotone and nonmonotone trust-region-based algorithms for large scale unconstrained optimization problems. Computational Optimization and Applications **54**(1), 27–43 (2013)
40. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent in function space. In: Proc. NIPS, vol. 12, pp. 512–518 (1999)
41. Moré, J.J.: Computing a trust region step. SIAM Journal on scientific and statistical computing **4**(3), 553–572 (1983)
42. Moré, J.J.: Trust regions and projected gradients. In: System Modelling and Optimization, pp. 1–13. Springer, New York, NY (1988)
43. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer New York (2006)
44. Powell, M.: A hybrid method for nonlinear equations. In: P. Robinowitz (ed.) Numerical Methods for Nonlinear Algebraic Equations, pp. 87–144. Gordon and Breach Science, London (1970)
45. Powell, M.: A new algorithm for unconstrained optimization. In: J.B. Rosen, O.L. Mangasarian, K. Ritter (eds.) Nonlinear Programming, pp. 31–66. Academic Press, New York (1970)
46. Powell, M.J.D.: Restart procedures for the conjugate gradient method. Mathematical programming **12**, 241–254 (1977)
47. Qi, L., Sun, J.: A nonsmooth version of newton's method. Mathematical Programming **58**, 353–367 (1993)
48. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Review **52**(3), 471–501 (2010). DOI 10.1137/070697835. URL https://doi.org/10.1137/070697835
49. Robinson, S.M.: Normal maps induced by linear transformations. Mathematics of Operations Research **17**(3), 691–714 (1992)
50. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM Journal on Numerical Analysis **20**(3), 626–637 (1983)

51. Sun, S., Nocedal, J.: A trust-region method for noisy unconstrained optimization. Mathematical Programming **202**, 445–472 (2023)
52. Toint, P.: Towards an efficient sparsity exploiting newton method for minimization. In: Sparse matrices and their uses, pp. 57–88. Academic press (1981)
53. Toint, P.L.: Global Convergence of a Class of Trust-Region Methods for Nonconvex Minimization in Hilbert Space. IMA Journal of Numerical Analysis **8**(2), 231–252 (1988). DOI 10.1093/imanum/8.2.231
54. Wang, X., Ma, S., Goldfarb, D., Liu, W.: Stochastic quasi-Newton methods for nonconvex stochastic optimization. SIAM Journal on Optimization **27**(2), 927–956 (2017)
55. Ziems, J.C., Ulbrich, S.: Adaptive multilevel inexact SQP methods for PDE-constrained optimization. SIAM Journal on Optimization **21**(1), 1–40 (2011). DOI 10.1137/080743160

A L^1-Specific Orthant-Based Method Subproblem Solver

The `OBM` subproblem solver in Section 6 is tailored to L^1-regularization and is adapted from the orthant-based method described in [13]. This solver has close ties to the subproblem solver described in [37] for linearly-constrained optimization. For this method, $X = L^2(D)$ defined on the measurable space $(D, \mathcal{F}, \mu)$ and

$$\phi(x) = \beta \|x\|_1 \coloneqq \beta \int_D |x| \, \mathrm{d}\mu,$$

for $\beta > 0$. Extending the notation in [1], we denote the minimum-norm subgradient of the model m_k at the ℓ-th subproblem iterate $x_{k,\ell}$ by

$$v_{k,\ell}(w) \coloneqq \begin{cases} g_{k,\ell}(w) + \beta & \text{if } x_{k,\ell}(w) > 0 \text{ or } (x_{k,\ell}(w) = 0 \wedge g_{k,\ell}(w) < -\beta) \\ g_{k,\ell}(w) - \beta & \text{if } x_{k,\ell}(w) < 0 \text{ or } (x_{k,\ell}(w) = 0 \wedge g_{k,\ell}(w) > \beta) \\ 0 & \text{if } x_{k,\ell}(w) = 0 \text{ and } g_{k,\ell}(w) \in [-\beta, \beta] \end{cases} \tag{49}$$

for $w \in D$, where $g_{k,\ell} \coloneqq \nabla f_k(x_{k,\ell})$. Note that $-v_{k,\ell}$ is the steepest descent direction for m_k at $x_{k,\ell}$ and the directional derivative $m_k'(x_{k,\ell}; -v_{k,\ell}) < 0$ whenever $-g_{k,\ell} \notin \partial\phi(x_{k,\ell})$, i.e.,

$$m_k'(x_{k,\ell}; -v_{k,\ell}) = \sup_{\eta \in \partial\phi(x_{k,\ell})} \left\langle g_{k,\ell} + \eta, -v_{k,\ell} \right\rangle = -\left\|v_{k,\ell}\right\|^2 < 0$$

[6, Prop. 17.22]. Using $v_{k,\ell}$, we define the active set $\mathcal{A}_{k,\ell} \coloneqq \{w \in D \mid v_{k,\ell}(w) = 0\}$. Roughly speaking, we eliminate the active components from the trust-region subproblem and only solve for the inactive ones $D \setminus \mathcal{A}_{k,\ell}$. Instead of computing a search direction $s_{k,\ell}$ by approximately solving the modified problem

$$\begin{aligned} &\min_{s \in X} \ \frac{1}{2} \langle B_k s, s \rangle + \langle v_{k,\ell}, s \rangle \\ &\text{subject to} \quad \|s\|_2 \le \Delta_k, \quad s(w) = 0 \quad \text{for a.a. } w \in \mathcal{A}_{k,\ell} \end{aligned} \tag{50}$$

using projected truncated CG [30], we compute $s_{k,\ell}$ by explicitly eliminating the active components. Let $P_{k,\ell} \in \mathcal{L}(X)$ denote the projection onto the inactive set $D \setminus \mathcal{A}_{k,\ell}$, i.e.,

$$[P_{k,\ell} s](w) \coloneqq s(w)(1 - \mathbb{1}_{\mathcal{A}_{k,\ell}}(w)),$$

where $\mathbb{1}_{\mathcal{A}_{k,\ell}}(w) = 1$ if $w \in \mathcal{A}_{k,\ell}$ and $\mathbb{1}_{\mathcal{A}_{k,\ell}}(w) = 0$ if $w \in D \setminus \mathcal{A}_{k,\ell}$. Then, we can rewrite (50) in reduced form as

$$\min_{s \in X} \ \frac{1}{2} \left\langle (P_{k,\ell}^* B_k P_{k,\ell}) s, s \right\rangle + \left\langle P_{k,\ell}^* v_{k,\ell}, s \right\rangle \quad \text{subject to} \quad \|P_{k,\ell} s\|_2 \le \Delta_k, \tag{51}$$

which we approximately solve using truncated CG [50]. Let $\hat{s}_{k,\ell} \in X$ denote an approximate solution to (51), then $s_{k,\ell} = P_{k,\ell} \hat{s}_{k,\ell}$ is an approximate solution to (50). Given $s_{k,\ell}$, we perform a backtracking line search to determine a step length that satisfies the sufficient decrease condition (17). The full routine is described in Algorithm 5.

Algorithm 5 Orthant-based subproblem solver for L^1-regularized problems

Require: The iteration limit $\texttt{maxit} \in \mathbb{N}$, decrease factor $\beta_{\text{dec}} \in (0,1)$, descent parameter $\mu \in (0,1)$, CG iteration limit $\texttt{maxiticg} \in \mathbb{N}$, and positive tolerances $\bar{\tau}$, τ_k, δ_{abs} and δ_{rel}

Set $\ell \leftarrow 0$ and compute the Cauchy point $x_{k,0} = x_k^c$
Compute $g_{k,0} \leftarrow g_k + B_k(x_{k,0} - x_k)$ and $h_{k,0} \leftarrow H(x_{k,0}, g_{k,0}, t_k)$
while $\ell < \texttt{maxit}$ **and** $h_{k,\ell} > \min\{\bar{\tau}, \tau_k h_{k,0}\}$ **and** $\|x_{k,\ell} - x_k\|_2 < \Delta_k$ **do**
  Compute $v_{k,\ell}$ from (49) and the corresponding active set $\mathcal{A}_{k,\ell}$
  Set $r \leftarrow P_{k,\ell}^* v_{k,\ell}$ and $\rho_1 \leftarrow \langle r, r \rangle$
  Set $d \leftarrow -r$ and $s_{k,\ell} \leftarrow 0$
  for $i = 1, \dots, \texttt{maxitcg}$ **do**
    Compute $b \leftarrow (P_{k,\ell}^* B_k P_{k,\ell}) d$ and $\kappa \leftarrow \langle b, d \rangle$
    if $\kappa \le 0$ **then**
      Compute $\alpha > 0$ as the solution to $\|x_{k,\ell} + s_{k,\ell} + \alpha d - x_k\| = \Delta_k$
      Set $s_{k,\ell} \leftarrow s_{k,\ell} + \alpha d$
      break
    end if
    Compute $\alpha \leftarrow \rho_i / \kappa$
    if $\|x_{k,\ell} + s_{k,\ell} + \alpha d - x_k\| \ge \Delta_k$ **then**
      Compute $\alpha > 0$ as the solution to $\|x_{k,\ell} + s_{k,\ell} + \alpha d - x_k\| = \Delta_k$
      Set $s_{k,\ell} \leftarrow s_{k,\ell} + \alpha d$
      break
    end if
    Update the step $s_{k,\ell} \leftarrow s_{k,\ell} + \alpha d$
    Update the residual $r \leftarrow r + \alpha b$
    Compute $\rho_{i+1} \leftarrow \langle r, r \rangle$
    if $\sqrt{\rho_{i+1}} \le \min\{\delta_{\text{abs}}, \delta_{\text{rel}} \sqrt{\rho_1}\}$ **then**
      break
    end if
    Compute $\beta \leftarrow \rho_{i+1} / \rho_i$
    Set the trial step $d \leftarrow \beta d - p$
  end for
  Set the step length $\sigma \leftarrow 1$
  Set the trial iterate $x_{k,\ell+1} \leftarrow x_{k,\ell} + \sigma s_{k,\ell}$
  while $m_k(x_{k,\ell+1}) > m_k(x_{k,\ell}) + \mu \min\{0, \langle g_{k,\ell}, x_{k,\ell+1} - x_{k,\ell} \rangle + \phi(x_{k,\ell+1}) - \phi(x_{k,\ell})\}$ **do**
    Set the step length $\sigma \leftarrow \beta_{\text{dec}} \sigma$
    Set the trial iterate $x_{k,\ell+1} \leftarrow x_{k,\ell} + \sigma s_{k,\ell}$
  end while
  Compute $g_{k,\ell+1} \leftarrow g_{k,\ell} + \sigma B_k s_{k,\ell}$ and $h_{k,\ell+1} \leftarrow H(x_{k,\ell+1}, g_{k,\ell+1}, t_k)$
  Update $\ell \leftarrow \ell + 1$
end while
Return $x_k^+ \leftarrow x_{k,\ell}$ as the approximate solution