Self-Correcting Variable-Metric Algorithms for Nonsmooth Optimization

Frank E. Curtis

Department of Industrial and Systems Engineering
Lehigh University, Bethlehem, PA, USA

Daniel P. Robinson

Department of Applied Mathematics and Statistics
Johns Hopkins University, Baltimore, MD, USA

Baoyu Zhou

Department of Industrial and Systems Engineering
Lehigh University, Bethlehem, PA, USA

COR@L Technical Report 17T-012

Self-Correcting Variable-Metric Algorithms for Nonsmooth Optimization*

Frank E. Curtis† Daniel P. Robinson‡ Baoyu Zhou§

October 18, 2017

Abstract

A generic algorithmic framework is proposed for minimizing nonsmooth and potentially nonconvex objective functions. The framework is variable-metric in the sense that, in each iteration, a step is computed using a symmetric positive definite matrix whose value is updated in a similar manner as in the Broyden-Fletcher-Goldfarb-Shanno (BFGS) scheme popular for minimizing smooth objectives. Unlike previously proposed variable-metric algorithms for minimizing nonsmooth functions, the proposed framework exploits the self-correcting properties of BFGS-type updating. In so doing, the framework does not overly restrict the manner in which the step computation matrices are updated, yet the scheme is controlled well enough that global convergence guarantees can be established. The results of numerical experiments for various algorithms are presented to demonstrate the self-correcting behaviors that are guaranteed by the framework.

1 Introduction

The purpose of this paper is to present an algorithmic framework for solving minimization problems involving nonsmooth objective functions. To frame the context and goals of this work, it is worthwhile to recall some history on the design of algorithms for nonlinear optimization over the past few decades.

Practical algorithms for minimizing smooth objective functions primarily fall between two extremes. At one extreme are steepest descent methods that only require first-order derivative (i.e., gradient) information of the objective function. Such methods have relatively cheap per-iteration costs and can attain a linear rate of convergence to a minimizer. At the other extreme are Newton methods that require first- and second-order derivative (i.e., respectively, gradient and Hessian) information as well as solutions of linear systems of equations of dimension equal to the number of variables. Such methods are relatively expensive, but can attain a quadratic rate of convergence to a minimizer. For further details, see, e.g., [2, 3, 15, 35, 41].

One often finds, however, that the most computationally efficient method for a given application does not follow either of these extremes. That is, one often finds that with only approximate second-order information, and with techniques that avoid expensive computations such as solves of linear systems of equations, one can better balance per-iteration costs with per-iteration improvement.

Along these lines, one of the most important developments for smooth optimization algorithms came with the advent of *variable-metric* algorithms in the 1960s [13]. This class of methods, which includes quasi-Newton methods such as those of the widely successful Broyden-Fletcher-Goldfarb-Shanno (BFGS) variety

*This material is based upon work supported by the U.S. Department of Energy under Award Number DE–SC0010615 and by the U.S. National Science Foundation under Award Numbers DMS–1319356 and CCF–1618717.

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA.
E-mail: frank.e.curtis@gmail.com.

‡Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA.
E-mail: daniel.p.robinson@gmail.com.

§Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA.
E-mail: baz216@lehigh.edu.

[5, 17, 18, 42], often offer an attractive alternative between extremes. Such methods only require first-order derivative information, can avoid the need to solve linear systems of equations, and yet can offer superlinear convergence rate guarantees [14].

Unfortunately, when it comes to minimizing nonsmooth objective functions, the array of available algorithms is more varied, and attempts to characterize and compare them run into various challenges. For example, the ideas underlying steepest descent, quasi-Newton, and Newton methodologies can all be extended for minimizing nonsmooth functions, but practical methods often involve computations beyond obtaining derivative information and solving linear systems, making the computational trade-offs between methods less straightforward. In addition, theoretical convergence rates for algorithms become more difficult to prove, meaning that one cannot rely so easily on such characterizations when comparing methods. For example, despite being introduced decades ago in the 1970s and being one of the most popular classes of methods for convex optimization, convergence rate guarantees for bundle methods have been limited [34] and only recently have bounds been proved (in special cases) that accurately capture the performance one often finds from them in practice [16].

All of this being said, many have observed that, as in smooth optimization, improved computational trade-offs between per-iteration cost and improvement are often attained by methods that employ both first-order derivative information and approximate second-order information. In this spirit, this paper proposes a new variable-metric algorithmic framework for solving nonsmooth optimization problems.

Variable-metric algorithms for nonsmooth optimization have previously been proposed [23, 30]. Broadly speaking, they can be grouped into three categories. First, there are techniques not built on quasi-Newton-type updating. An important example in this group is Shor's R-algorithm [24, 43]. Second, there are techniques that attempt to employ quasi-Newton ideas, but only possess convergence guarantees when the updates are restricted to ensure that the resulting Hessian approximations remain sufficiently positive definite and bounded in all iterations [11, 12]. Third, there are techniques that employ unadulterated quasi-Newton ideas. Interestingly, convergence guarantees can be established for such methods in a few specific cases [31], though general guarantees for broad classes of functions remain elusive.

The algorithmic framework proposed in this paper falls into the second of these categories of methods, but is unique in that it attempts to exploit the *self-correcting properties* of BFGS-type updating. These properties guarantee that a sufficient number of matrices generated by a BFGS-type updating scheme possess useful properties for ensuring convergence without having to overly restrict the manner in which the updates are performed. The hope is that the framework proposed in this paper can offer both practical performance gains for various algorithm classes for nonsmooth optimization as well as outline how these useful properties of BFGS-type updating can be incorporated into other algorithms. One interesting behavior of our approach, revealed by our numerical experiments, is that it is effective at recognizing when the current iterate is nearly stationary. This is in contrast to the behavior of a straightforward BFGS-based approach, which, if employed in practice, might break down (due to a failed line search) before any stationarity guarantee has been offered.

1.1 Organization

The paper is organized as follows. In §2, we formally state our problem of interest, describe the proposed algorithmic framework, and discuss at a broad level the types of algorithms that adhere to the framework. In §3, we discuss the properties of the scaling matrices employed in the framework, then show how these properties can be used to obtain generic convergence guarantees for the framework. In §4, we present a specific algorithm that adheres to the framework. The results of numerical experiments for this algorithm as well as others that are inspired by our framework are given in §5. Concluding remarks are provided in §6.

1.2 Notation

Let $\mathbb{R}$ denote the set of real numbers (i.e., scalars), let $\mathbb{R}_+$ denote the set of nonnegative real numbers, let $\mathbb{R}_{++}$ denote the set of positive real numbers, and let $\mathbb{N} := \{1, 2, \dots\}$ denote the set of natural numbers. In addition, for any of these quantities, let a superscript $n \in \mathbb{N}$ be used to indicate the n-dimensional extension of the set—e.g., let $\mathbb{R}^n$ denote the set of n-dimensional real vectors—and let a superscript $\overline{n} \times n$

with $(\overline{n}, n) \in \mathbb{N} \times \mathbb{N}$ be used to indicate the $\overline{n}$-by-n-dimensional extension of the set—e.g., let $\mathbb{R}^{\overline{n}\times n}$ denote the set of $\overline{n}$-by-n real matrices. A vector with all elements equal to 1 is denoted as $\mathbb{1}$ and an identity matrix is denoted as I, where, in each case, the size of the quantity is determined by the context in which it appears. With real symmetric matrices A and B, let $A \succ (\succeq)\ B$ indicate that $A - B$ is positive definite (semidefinite). Given a set $\mathcal{X}$, its convex hull is denoted as $\operatorname{conv} \mathcal{X}$.

2 Problem and Algorithmic Framework

In this section, we formally state our optimization problem of interest and our proposed algorithmic framework. We also outline ideas underlying various types of algorithms that adhere to the proposed framework. The complete details of a specific instance of the framework is left until §4 after generic convergence results are motivated and proved in §3.

2.1 Problem Statement

Our problem of interest is to minimize an objective function $f : \mathbb{R}^n \to \mathbb{R}$, i.e., consider the optimization problem

$$\min_{x\in\mathbb{R}^n} \ f(x). \tag{P}$$

For now, only the following assumption is made about problem (P).

Assumption 2.1. *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ in the optimization problem (P) is bounded below over $\mathbb{R}^n$, locally Lipschitz on $\mathbb{R}^n$, and continuously differentiable in an open dense set $\mathcal{D}$ with full measure in $\mathbb{R}^n$.*

This assumption represents one of the loosest sets of conditions under which one can guarantee convergence by popular algorithms, such as bundle [23, 26, 34, 41] and gradient sampling [6, 29] methods. Under it, there exists a scalar $f_{\text{inf}} \in \mathbb{R}$ such that

$$f(x) \geq f_{\text{inf}} \quad \text{for all} \quad x \in \mathbb{R}^n, \tag{2.1}$$

and, for any compact subset $\mathcal{B}$ of $\mathbb{R}^n$, there exists a constant $L_{\mathcal{B}} \in \mathbb{R}_{++}$ such that

$$|f(x) - f(\bar{x})| \leq L_{\mathcal{B}}\|x - \bar{x}\|_2 \quad \text{for all} \quad (x, \bar{x}) \in \mathcal{B} \times \mathcal{B}. \tag{2.2}$$

However, Assumption 2.1 does not preclude the possibility that f might have no minimizer, or that it might have many local minimizers. Since our proposed algorithmic framework employs local search techniques and is not intended to search for a global minimizer of f, it is worthwhile to derive stationarity conditions for f that must be satisfied at any local minimizer. Overall, the goal of the proposed framework is to characterize a family of methods for generating a sequence of iterates that is guaranteed, in the limit, to reveal a stationary point for f.

Remark 2.2. *One might be interested in situations when the objective function can be unbounded below and/or when it is extended-real-valued, i.e., when $f : \mathbb{R}^n \to (\mathbb{R} \cup \{-\infty, \infty\})$. We claim that the proposed framework, which ensures monotonic decrease in the objective function, is also viable in such cases, at least as long as one has access to an initial iterate x_1 in the effective domain of f, i.e., $x_1 \in \operatorname{dom}(f) := \{x \in \mathbb{R}^n : f(x) < \infty\}$. If f is unbounded below and an iterate sequence $\{x_k\}$ is generated such that $\{f(x_k)\} \searrow -\infty$, then there is nothing else that one should ask from the proposed framework. Hence, for simplicity, Assumption 2.1 precludes this case by ensuring that any such sequence $\{f(x_k)\}$ is bounded below. As for cases when the objective function is extended-real-valued, we claim that if any stationary point for f lies in the interior of the effective domain $\operatorname{dom}(f)$, then, with slight modifications of the proposed framework—e.g., to handle points encountered outside of $\operatorname{dom}(f)$—the analysis presented for the proposed framework follows in essentially the same manner as under Assumption 2.1. Moreover, even if a stationary point lies on the relative boundary of $\operatorname{dom}(f)$, we claim that the proposed framework can still be viable with suitable modifications—again, for handling points encountered not inside of $\operatorname{dom}(f)$.*

Stationarity conditions for f can be derived following the treatment by Clarke [9]. (Indeed, many of the following terms are often defined with a "Clarke" designation. However, for brevity, we omit this designation throughtout the paper.) First, the generalized directional derivative of f at $x \in \mathbb{R}^n$ with respect to $s \in \mathbb{R}^n$ is given by

$$f^\circ(x;s) = \limsup_{\bar{x} \to x, \alpha \searrow 0} \frac{f(\bar{x}+\alpha s) - f(\bar{x})}{\alpha}.$$

The subdifferential of f at x is then defined as

$$\partial f(x) = \{g \in \mathbb{R}^n : f^\circ(x;s) \geq g^T s \text{ for all } s \in \mathbb{R}^n\}.$$

According to Rademacher's theorem, any function f that is locally Lipschitz on $\mathbb{R}^n$ is differentiable almost everywhere and its subdifferential at x satisfies (see [9, Th. 2.5.1])

$$\partial f(x) = \operatorname{conv}\left\{\lim_{k\to\infty} \nabla f(x_k) : \{x_k\} \to x \text{ and } \{x_k\} \subset \mathcal{D}\right\}.$$

For a given $\epsilon \in \mathbb{R}_+$, the ϵ-subdifferential [19] of f at x is given by

$$\partial_\epsilon f(x) = \operatorname{conv} \partial f(\mathbb{B}(x,\epsilon)), \quad \text{where} \quad \mathbb{B}(x,\epsilon) := \{\bar{x} \in \mathbb{R}^n : \|\bar{x} - x\|_2 \leq \epsilon\}. \tag{2.3}$$

A point $x \in \mathbb{R}^n$ is said to be stationary for f if $0 \in \partial f(x)$ whereas it is merely ϵ-stationary for $\epsilon \in \mathbb{R}_+$ if $0 \in \partial_\epsilon f(x)$. The following result will be used later on.

Lemma 2.3. ([29, Lemma 2.3(iii)]) *Let $\{x_k\} \subset \mathbb{R}^n$ and $\{\epsilon_k\} \subset \mathbb{R}_+$ be infinite sequences and define $\{\tilde{g}_k\} \subset \mathbb{R}^n$ such that $\tilde{g}_k \in \partial_{\epsilon_k} f(x_k)$ for all $k \in \mathbb{N}$. If, for $x \in \mathbb{R}^n$,*

$$\liminf_{k\to\infty} \max\{\|x_k - x\|_2, \|\tilde{g}_k\|_2, \epsilon_k\} = 0,$$

then $0 \in \partial f(x)$, i.e., the point x is stationary for f.

2.2 Algorithmic Framework

The framework that we propose, entitled a ***S**elf-correcting **V**ariable-metric **A**lgorithm for **N**onsmooth **O**ptimization*, is stated below as the SVANO Framework. The framework consists of two main procedures: (*i*) Steps 2–4, the computation of a step yielding a reduction in the objective function and (*ii*) Steps 5–7, the computation of quantities used to update a scaling matrix to be used in the step computation procedure in the subsequent iteration. In SVANO, these procedures are written in a generic manner so as to allow for flexibility in the choices of various algorithmic quantities. Specific techniques for choosing these quantities are given in the following subsection, and specific conditions that these quantities must satisfy in order to ensure convergence are described along with the analysis in §3.

The step computation procedure in SVANO covers a wide range of techniques employed in the nonsmooth optimization literature, including those that employ cutting plane and gradient sampling methodologies using line search and/or trust region techniques. Given a symmetric positive definite matrix W_k (i.e., given $W_k \succ 0$), the procedure consists of the selection of a set of points $\{x_{k,j}\}_{j=1}^m$ in the vicinity of (and including) the current iterate $x_k \in \mathbb{R}^n$ and a set of vectors $\{g_{k,j}\}_{j=1}^m$ where $g_{k,j}$ for each $(k,j) \in \mathbb{N} \times \{1,\ldots,m\}$ represents a convex combination of subgradients of f evaluated at points in a vicinity of $x_{k,j}$. Following the selection of these vectors, the framework requires a pair (ω_k, γ_k) such that the step s_k in (2.5) leads to the reduction in f in (2.7). The vector ω_k, required to be nonnegative with elements summing to unity, should be viewed as a vector of weights such that, with G_k defined in (2.4), the step component $G_k\omega_k$ is a convex combination of the elements in the set $\{g_{k,j}\}_{j=1}^m$. The vector γ_k then represents a perturbation of this convex combination, which, e.g., may arise due to the use of line search or trust region methodologies; see §2.3.

A critical feature of SVANO is that each element of the sequence of matrices $\{W_k\}_{k\geq 2}$ is set by an update performed during the previous iteration. The update (2.10) has the same form as a standard BFGS update

SVANO

Require: A matrix $\overline{H} \succ 0$ with $(\lambda_{\min}, \lambda_{\max}) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that

$$\lambda_{\min}\|s\|_2^2 \leq s^T \overline{H} s \leq \lambda_{\max}\|s\|_2^2 \quad \text{for all} \quad s \in \mathbb{R}^n;$$

parameters $\alpha \in (0,1)$, $\eta \in (0, \lambda_{\min}]$, and $\theta \in [\lambda_{\max}, \infty)$; an initial point $x_1 \in \mathbb{R}^n$; and an initial positive definite inverse Hessian approximation $W_1 \in \mathbb{R}^{n \times n}$.

1: **for** $k \in \mathbb{N}$ **do**

2: Compute, for some $m \in \mathbb{N}$,

$$\{x_{k,j}\}_{j=1}^m \subset \mathbb{R}^n \text{ with } x_{k,1} \leftarrow x_k,$$
$$\{g_{k,j}\}_{j=1}^m \subset \mathbb{R}^n,$$
$$\omega_k \in \mathbb{R}_+^m \text{ with } \mathbb{1}^T \omega_k = 1,$$
$$\text{and} \quad \gamma_k \in \mathbb{R}^n$$

3: such that setting

$$G_k \leftarrow \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix}, \tag{2.4}$$
$$s_k \leftarrow -W_k(G_k\omega_k + \gamma_k), \tag{2.5}$$
$$\text{and} \quad x_{k+1} \leftarrow x_k + s_k \tag{2.6}$$

4: yields

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\alpha(G_k\omega_k + \gamma_k)^T W_k (G_k\omega_k + \gamma_k). \tag{2.7}$$

5: Choose $y_k \in \mathbb{R}^n$ and compute β_k as the smallest value in $[0,1]$ such that

$$v_k \leftarrow \beta_k \overline{H} s_k + (1 - \beta_k) y_k \tag{2.8}$$

6: yields

$$\eta \leq \frac{s_k^T v_k}{\|s_k\|_2^2} \quad \text{and} \quad \frac{\|v_k\|_2^2}{s_k^T v_k} \leq \theta, \tag{2.9}$$

7: then set

$$W_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T W_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}. \tag{2.10}$$

8: **end for**

from the smooth optimization literature, and, indeed, the framework is designed to exploit the properties induced by such an update. However, it is important to note that the framework allows flexibility in the choice of y_k—in theory, any element of $\mathbb{R}^n$ will suffice—as long as the scalar $\beta_k \in [0,1]$ is chosen such that the bounds in (2.9) are satisfied; see [10] for the introduction of this idea for stochastic optimization. One possible choice for y_k is the displacement between a subgradient of f at x_{k+1} with one at x_k, which may seem natural since this is the choice that can lead to local superlinear convergence guarantees when the objective function f is smooth. However, given that SVANO is designed to solve nonsmooth problems, one should not consider this as the only reasonable choice for y_k. (For example, one might choose the difference between convex combinations of subgradients encountered in the consecutive iterations.) An important conclusion of the analysis in §3 is that the bounds (2.9) are sufficient for ensuring convergence guarantees for the framework, and these bounds can be satisfied for any y_k as long as v_k is chosen in (2.8) with sufficiently large $\beta_k \in [0,1]$. (The allowed ranges for the parameters η and θ are set so that (2.9) is well-defined. In particular, note that with $\beta_k = 1$ one obtains $v_k = \overline{H} s_k$ for which, due to the ranges for η and θ, it follows that (2.9) holds; hence, $\beta_k \in [0,1]$ always exists such that v_k set by (2.8) satisfies (2.9).) We discuss various choices for the matrix $\overline{H} \succ 0$ and the sequence $\{y_k\}$ along with the description of our software implementation in §5.

As is well known, applying the Sherman-Morrison-Woodbury formula to (2.10) yields the following updating formula for $\{H_k\}$ where $H_k = W_k^{-1}$ for all $k \in \mathbb{N}$:

$$H_{k+1} \leftarrow \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right)^T H_k \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right) + \frac{v_k v_k^T}{s_k^T v_k}. \tag{2.11}$$

Despite the fact that SVANO does not require $\{H_k\}$ explicitly, it is useful to define this sequence for various reasons. For one thing, it can be observed from (2.5) that the sufficient reduction condition (2.7) can equivalently be written as

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2}\alpha s_k^T H_k s_k, \tag{2.12}$$

i.e., the condition requires that the reduction in f yielded from x_k to x_{k+1} is proportional to a quadratic function of the step s_k, which is a typical requirement in nonlinear optimization algorithms. In addition, the properties of the sequence $\{H_k\}$ corresponding to $\{W_k\}$ will be of central importance in the analysis in §3.

It is worthwhile to mention that (2.8) with $\overline{H} \equiv H_k$ would reflect a standard *damping* of the BFGS update [38, 35]. However, rather than employ an element of the *sequence* $\{H_k\}$ in (2.8), we employ the *fixed* matrix $\overline{H}$. Since this allows us to ensure that (2.9) holds for the constants η and θ for all $k \in \mathbb{N}$, we are able to ensure the self-correcting properties that are central to our convergence analysis. One cannot maintain such assurances if H_k is used in place of $\overline{H}$ in (2.8). Another alternative would be to employ a sequence $\{\overline{H}_k\}$ with eigenvalues uniformly bounded below by $\lambda_{\min}$ and above by $\lambda_{\max}$. That said, for simplicity, let us assume that $\overline{H}$ is fixed.

For ease of reference throughout the remainder of the paper, we refer to $\{H_k\}$ and $\{W_k\}$ as sequences of *Hessian approximations* and *inverse Hessian approximations*, respectively. This terminology should be easy to accept since it is common in the literature on quasi-Newton methods, even for nonsmooth optimization. However, since f is nonsmooth, the term "Hessian" should be taken loosely as a matrix that approximates changes in the subgradients of f taken at nearby points in $\mathbb{R}^n$. See [9] for more information about generalized second derivatives for nonsmooth functions.

2.3 Step Computation Techniques

The step computation procedure in the SVANO Framework encapsulates many techniques proposed in the literature. To justify this claim, suppose that at an iterate $x_k \in \mathbb{R}^n$, a set of points $\{x_{k,j}\}_{j=1}^m$ (with $x_{k,1} \leftarrow x_k$) and vectors $\{g_{k,j}\}_{j=1}^m \subset \mathbb{R}^n$ are given as described in the framework. In addition, suppose that a set of scalars $\{f_{k,j}\}_{j=1}^m$ are given corresponding to evaluations of f at $\{x_{k,j}\}_{j=1}^m$ or other points that have been encountered. Then, a convex piecewise-linear model of f at x_k is given by $l_{k,m} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$l_{k,m}(x) = \max_{j \in \{1,\ldots,m\}} \{f_{k,j} + g_{k,j}^T (x - x_{k,j})\}. \tag{2.13}$$

Also given a positive definite Hessian approximation $H_k \succ 0$, a convex piecewise-quadratic model of f at x_k is given by $q_{k,m} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$q_{k,m}(x) = l_{k,m}(x) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k). \tag{2.14}$$

A step toward minimizing f can be defined by the minimizer of $q_{k,m}$ within a region defined by a norm $\|\cdot\|$ and trust region radius $\delta_k \in \mathbb{R}_{++} \cup \{\infty\}$, i.e., the minimizer of

$$\min_{x \in \mathbb{R}^n} \quad q_{k,m}(x) \quad \text{s.t.} \quad \|x - x_k\| \leq \delta_k. \tag{2.15}$$

Solving (2.15) directly can be challenging due to the nonsmoothness of $l_{k,m}$ (and, hence, of $q_{k,m}$) and due to the presence of the trust region constraint (if $\delta_k < \infty$). One can reformulate it as the smooth constrained quadratic optimization problem (QP)

$$\begin{aligned} \min_{(x,z) \in \mathbb{R}^n \times \mathbb{R}} \quad & z + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \\ \text{s.t.} \quad & f_{k,j} + g_{k,j}^T (x - x_{k,j}) \leq z \quad \text{for all} \quad j \in \{1, \ldots, m\} \quad \text{and} \\ & x \in \mathcal{X}_k := \{x \in \mathbb{R}^n : \|x - x_k\| \leq \delta_k\}, \end{aligned} \tag{2.16}$$

but even this formulation can be difficult to solve to high accuracy. The dual of this subproblem, on the other hand, has attractive properties. Introducing Lagrange multipliers $\{\omega_j\}_{j=1}^m$ that form $\omega \in \mathbb{R}^m$ and denoting the dual norm of $\|\cdot\|$ as $\|\cdot\|_*$, the dual of (2.16) can be simplified (see Appendix B) to

$$\sup_{(\omega,\gamma)\in\mathbb{R}^m_+\times\mathbb{R}^n} -\tfrac{1}{2}(G_k\omega+\gamma)^T W_k(G_k\omega+\gamma) + b_k^T\omega - \delta_k\|\gamma\|_* \quad \text{s.t.} \quad \mathbb{1}^T\omega = 1, \tag{2.17}$$

where the vector $b_k \in \mathbb{R}^m$ has as its jth component

$$b_{k,j} = f_{k,j} + g_{k,j}^T(x_k - x_{k,j}). \tag{2.18}$$

The constraints of this dual merely involve a single affine equality constraint and lower bounds on the dual variables. In addition, when $\delta_k = \infty$, then $\gamma_k = 0$ and the dual objective is a smooth quadratic and, for the resulting QP, specialized solvers exist; e.g., see [11, 27]. Such an algorithm for solving the dual subproblem (2.17) is likely to be more efficient than one for minimizing the nonsmooth function in (2.15) or the equivalent smooth constrained QP (2.16). In any case, if one solves the dual (2.17), then the solution to (2.15) can be recovered from that of (2.17), as stated as part of the following lemma; for a complete proof of this result, see Appendix B.

Lemma 2.4. *Given the solution $(\omega_k, \gamma_k) \in \mathbb{R}^m_+ \times \mathbb{R}^n$ of the dual subproblem (2.17), the solution of the primal subproblem (2.15) is given by $x_k - W_k(G_k\omega_k + \gamma_k)$; hence, x_{k+1} in (2.6) with s_k given in (2.5) is the solution of (2.15). In addition,*

$$f(x_k) - l_{k,m}(x_{k+1}) \geq \tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k), \tag{2.19}$$

meaning that, if

$$f(x_k) - f(x_{k+1}) \geq \alpha(f(x_k) - l_{k,m}(x_{k+1})), \tag{2.20}$$

then (2.7) holds. Finally, it also holds that

$$f(x_k) - l_{k,m}(x_{k+1}) \geq \tfrac{1}{2}\omega_k^T G_k^T W_k G_k\omega_k - \tfrac{1}{2}\gamma_k^T W_k\gamma_k - \|\gamma_k\|_*. \tag{2.21}$$

There are also practical benefits of solving the dual problem (2.17) when $\{x_{k,j}\}_{j=1}^m$ and corresponding quantities are generated incrementally. For example, suppose that elements indexed by $j \in \{1,\ldots,\overline{m}\}$ for some $\overline{m} \in \mathbb{N}$ have been generated, but the resulting trial iterate defined as in (2.6) fails to satisfy (2.7). Then, suppose that additional data indexed by $j = \overline{m}+1$ is generated in some manner to produce the next trial iterate. The dual subproblem is the same as the previous one, except for the addition of a single dual variable (and corresponding objective and constraint data entries). The previous optimal dual solution augmented with the new variable initialized to zero represents a feasible solution of the subsequent dual problem, making it an attractive starting point for the solve of the subsequent dual subproblem.

3 Convergence of SVANO

In this section, we explore properties of *any* sequences $\{W_k\}$ and $\{H_k\}$ generated by (2.10) and (2.11), respectively, then discuss generic convergence properties of the SVANO Framework. To start, one finds that the updates (2.10) and (2.11) satisfy *secant-like* equations, namely

$$W_{k+1}v_k = s_k \quad \text{and} \quad H_{k+1}s_k = v_k. \tag{3.1}$$

One can also derive a geometric interpretation of the updates for the Hessian approximations, revealing that the kth update can be viewed as the combination of a *projection* to erase curvature information along s_k—in a sense, temporarily setting $s_k^T H_{k+1} s_k$ to zero—along with a *correction* of said curvature based on information contained in v_k to yield $s_k^T H_{k+1} s_k = s_k^T v_k > 0$; see Appendix A. Most importantly for our purposes is that one can show that sequences of such updates result in useful self-correcting properties, as explored below in §3.1. These self-correcting properties of the Hessian approximations, when cast in terms of the inverse Hessian approximations, yield properties that we use to prove a convergence result for SVANO in §3.2.

3.1 Self-Correcting Properties of BFGS Updating

It is illustrated in Appendix A that the update (2.10) is a combination of a projection and a correction of the corresponding Hessian approximation. However, as these updates build upon one another from one iteration to the next, it is important to characterize properties of the resulting matrices and their effects on the computed steps after a *sequence* of updates have been performed. The known fact that we state in this subsection is that as long as v_k is chosen to satisfy the two critical inequalities in (2.9), then despite the projections that each project out curvature information along $\operatorname{span}(s_k)$ with each update, the corrections will be sufficient to ensure that the sequences of Hessian and inverse Hessian approximations satisfy useful inequalities.

It is worthwhile to note that early work on the convergence of quasi-Newton methods by Powell [37] and others [8, 39, 40, 44] involved performing analyses that bound the growth of the traces and the determinants of $\{H_k\}$. In what follows, we follow the work in [7] involving a streamlined approach in which one bounds the growth of a function defined by a combination of these quantities; see also the summary provided in [35]. A similar review of these results was provided in [10].

Given $H \succ 0$, consider the function $\psi : \mathbb{R}^{n\times n} \to \mathbb{R}$ defined by

$$\psi(H) = \operatorname{trace}(H) - \ln(\det(H)).$$

It can be shown that $\psi(H)$ is positive (in fact, at least n) and represents a measure of closeness between H and the identity matrix I (for which $\psi(I) = n$); in particular, $\psi(H)$ is an upper bound for the natural logarithm of the condition number of H. In addition, the update (2.11) implies that, for all $k \in \mathbb{N}$, one has

$$\operatorname{trace}(H_{k+1}) = \operatorname{trace}(H_k) - \frac{\|H_k s_k\|_2^2}{s_k^T H_k s_k} + \frac{\|v_k\|_2^2}{s_k^T v_k} \tag{3.2a}$$

$$\text{and (see [36])} \quad \det(H_{k+1}) = \det(H_k)\left(\frac{s_k^T v_k}{s_k^T H_k s_k}\right), \tag{3.2b}$$

with which one can explicitly relate $\psi(H_{k+1})$ and $\psi(H_k)$. Specifically, assuming that $H_k \succ 0$ and the iterate displacement satisfies $s_k \neq 0$, then by defining

$$\cos\phi_k := \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \text{and} \quad \iota_k := \frac{s_k^T H_k s_k}{\|s_k\|_2^2} \tag{3.3}$$

it follows from (3.2) that

$$\begin{aligned}\psi(H_{k+1}) = \psi(H_k) &+ \underbrace{\frac{\|v_k\|_2^2}{s_k^T v_k} - 1 - \ln\left(\frac{s_k^T v_k}{\|s_k\|_2^2}\right)}_{\in\mathbb{R}} \\ &+ \underbrace{\ln(\cos^2\phi_k)}_{\leq 0} + \underbrace{\left(1 - \frac{\iota_k}{\cos^2\phi_k} + \ln\left(\frac{\iota_k}{\cos^2\phi_k}\right)\right)}_{\leq 0}.\end{aligned} \tag{3.4}$$

Nonpositivity of the latter two terms is easily verified; see Appendix C.

By restricting the growth of ψ over $\{H_k\}$ and noting that there must exist certain iterations in which the latter terms in (3.4) are not too negative, one can prove the following theorem showing *self-correcting properties* of the update (2.11). For completeness, we provide a proof of this theorem in Appendix C; see also [7, Thm. 2.1].

Theorem 3.1. *Let the sequence of Hessian approximations $\{H_k\}$ satisfy (2.11) and suppose that there exist $(\eta, \theta) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that (2.9) holds for all $k \in \mathbb{N}$. Then, for any $p \in (0, 1)$, there exist constants $(\kappa, \sigma, \mu) \in \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that, for any $K \in \{2, 3, \dots\}$, the following hold for at least $\lceil pK \rceil$ values of $k \in \{1, \dots, K\}$:*

$$\kappa \leq \frac{s_k^T H_k s_k}{\|s_k\|_2 \|H_k s_k\|_2} \quad \text{and} \quad \sigma \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \mu. \tag{3.5}$$

This theorem leads to the following corollary about the inverse approximations.

Corollary 3.2. *Let the sequence of inverse Hessian approximations $\{W_k\}$ satisfy (2.10) and suppose that there exist $(\eta, \theta) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that (2.9) holds for all $k \in \mathbb{N}$. Then, for any $p \in (0,1)$, there exist constants $(\nu, \xi) \subset \mathbb{R}_{++}$ such that, for any $K \in \{2, 3, \dots\}$, the following hold for at least $\lceil pK \rceil$ values of $k \in \{1, \dots, K\}$:*

$$\begin{aligned} \nu \|G_k \omega_k + \gamma_k\|_2^2 \leq (G_k \omega_k + \gamma_k)^T W_k (G_k \omega_k + \gamma_k) \quad \text{and} \\ \|W_k (G_k \omega_k + \gamma_k)\|_2^2 \leq \xi \|G_k \omega_k + \gamma_k\|_2^2. \end{aligned} \tag{3.6}$$

Proof. The conditions of Theorem 3.1 hold. Hence, by (2.5) and since the inverse Hessian approximations satisfy $W_k = H_k^{-1}$ for all $k \in \mathbb{N}$, the inequalities in (3.5) can be rewritten using the notation $\bar{g}_k := G_k \omega_k + \gamma_k$ as

$$\kappa \leq \frac{\bar{g}_k^T W_k \bar{g}_k}{\|W_k \bar{g}_k\|_2 \|\bar{g}_k\|_2} \quad \text{and} \quad \sigma \leq \frac{\|\bar{g}_k\|_2}{\|W_k \bar{g}_k\|_2} \leq \mu. \tag{3.7}$$

From the first and third of the inequalities in (3.7), it follows that

$$\bar{g}_k^T W_k \bar{g}_k \geq \kappa \|W_k \bar{g}_k\|_2 \|\bar{g}_k\|_2 \geq (\kappa/\mu) \|\bar{g}_k\|_2^2,$$

so that the first inequality in (3.6) holds with $\nu := \kappa/\mu$. Meanwhile, from the second inequality in (3.7), it follows that $\|W_k \bar{g}_k\|_2^2 \leq \sigma^{-2} \|\bar{g}_k\|_2^2$, so that the second inequality in (3.6) holds with $\xi := \sigma^{-2}$, as claimed. □

3.2 Convergence of the SVANO Framework

In this subsection, we provide a few convergence results for the SVANO Framework. Our goal is merely to prove generic results that are useful in various circumstances. In §4, we present a specific algorithm instance that offers the guarantees presented here.

Our first result represents a fundamental component of the convergence theory for any algorithm that falls under the SVANO Framework. In particular, it shows that with the self-correcting properties of the employed BFGS-type updating scheme, there exists an infinite subsequence of iterations for which sufficient decreases in the objective f guarantee that subsequences of $\{G_k \omega_k + \gamma_k\}$ and $\{s_k\}$ vanish.

Theorem 3.3. *There exists an infinite index set $\mathcal{K} \subseteq \mathbb{N}$ such that the sequences $\{(G_k, \omega_k, \gamma_k)\}$ and $\{s_k\}$ computed by SVANO respectively satisfy*

$$\lim_{k \in \mathcal{K}, k \to \infty} \|G_k \omega_k + \gamma_k\|_2 = 0 \quad \text{and} \quad \lim_{k \in \mathcal{K}, k \to \infty} \|s_k\|_2 = 0. \tag{3.8}$$

Proof. It follows by the condition in Step 7 that, for all $k \in \mathbb{N}$, one has

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2} \alpha (G_k \omega_k + \gamma_k)^T W_k (G_k \omega_k + \gamma_k). \tag{3.9}$$

For a given $p \in (0,1)$, let $\mathcal{K} \subseteq \mathbb{N}$ be the infinite set of indices for which Corollary 3.2 guarantees the existence of $(\nu, \xi) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$ such that (3.6) holds for all $k \in \mathcal{K}$. Then, for all $k \in \mathcal{K}$, it follows from (3.9) and the first inequality in (3.6) that

$$f(x_{k+1}) \leq f(x_k) - \tfrac{1}{2} \nu \alpha \|G_k \omega_k + \gamma_k\|_2^2.$$

Since f is bounded below (see (2.1)) and monotonically decreasing, the first limit in (3.8) holds. This limit may be combined with the second inequality in (3.6) and the choice of s_k in (2.5) to obtain the second limit in (3.8). □

The conclusions of Theorem 3.3 are not entirely consequential in their own right. However, the theorem is fundamental in that it can be used to show that if the columns of G_k correspond to (combinations of) subgradients of f evaluated at points in the vicinity of x_k for all $k \in \mathbb{N}$, then, as long as the vanishing of $\{G_k \omega_k + \gamma_k\}$ implies the vanishing of $\{G_k \omega_k\}$ (at least over a subsequence), the first limit in (3.8) must mean that a stationary point of f is revealed by a subsequence of the iterates.

Theorem 3.4. *Suppose that the infinite index set $\mathcal{K} \subseteq \mathbb{N}$ in Theorem 3.3 admits an infinite subset $\mathcal{K}' \subseteq \mathcal{K}$ such that*

$$\lim_{k \in \mathcal{K}', k \to \infty} \|G_k \omega_k\|_2 = 0. \tag{3.10}$$

In addition, suppose that for all $k \in \mathcal{K}'$ there exists $\epsilon_k \in \mathbb{R}_{++}$ such that, for all $j \in \{1, \ldots, m\}$, the vector $g_{k,j}$ is a convex combination of subgradients of f evaluated at a finite subset of $\mathbb{B}(x_k, \epsilon_k)$ as defined in (2.3). Then, if for some $x \in \mathbb{R}^n$ one has

$$\liminf_{k \in \mathcal{K}', k \to \infty} \max\{\|x_k - x\|_2, \epsilon_k\} = 0, \tag{3.11}$$

then $0 \in \partial f(x)$, i.e., the limit point x of $\{x_k\}_{k \in \mathcal{K}'}$ is stationary for f.

Proof. Under the stated conditions, it follows that $g_{k,j} \in \partial_{\epsilon_k} f(x_k)$ for all $(k, j) \in \mathcal{K}' \times \{1, \ldots m\}$. Since $\partial_{\epsilon_k} f(x_k)$ is convex by its definition and $\tilde{g}_k := G_k \omega_k$ is a convex combination of $\{g_{k,j}\}_{j=1}^m$, it follows that $\tilde{g}_k \in \partial_{\epsilon_k} f(x_k)$ for all $k \in \mathcal{K}'$. Combining this with (3.10) and (3.11), the result follows from Lemma 2.3. □

The previous theorem reveals useful consequences of the first limit in (3.8). What, if any, are useful consequences of the second limit? These depend on the instance of SVANO of interest. For instances where, perhaps under additional assumptions about the objective function f, one can guarantee the existence of $\{\epsilon_k\}_{k \in \mathcal{K}'}$ and $x \in \mathbb{R}^n$ such that (3.11) holds, the second limit is not of great interest in itself. However, for other instances, having the subsequence of step norms converging to zero helps to ensure that (3.11) holds for some $\{\epsilon_k\}_{k \in \mathcal{K}'}$ and $x \in \mathbb{R}^n$. This is demonstrated for our specific instance of the framework considered in the next section.

4 An Instance of SVANO

Our primary goal in this section is to present a particular instance of SVANO that yields the convergence guarantee in Theorem 3.4. However, before doing so, we believe it is instructive to show how, if one tries to fit a classical BFGS strategy into the framework, certain behaviors might cause the method to falter. This helps to motivate the more involved strategy that we present.

4.1 Classical BFGS method

Let us follow [31] and discuss a BFGS method with a weak Wolfe line search. A description of a step computation procedure for such an algorithm is presented as BFGS-Step below, which should be viewed as a specific instance of the step computation written generically as Steps 2–4 in SVANO. Rather than delineate the details of a weak Wolfe line search, we direct the reader to [31, Alg. 4.6] and note that our required condition (4.1) corresponds to "c_1"$= \frac{1}{2}\alpha$, "s"$= g_{k,1}^T \tilde{s}_k$, and "t"$= \tilde{\alpha}_k^2$, which with the subsequent choices for s_k and x_{k+1} implies that (4.1) yields (2.7) with $G_k \omega_k = g_{k,1}$ and $\gamma_k = (\tilde{\alpha}_k - 1) g_{k,1}$. When the line search fails to produce a sufficiently large stepsize, the algorithm *breaks down* [31, §6.1].

The issues that may arise for this algorithm all relate to the line search. If for some $k \in \mathbb{N}$ the function f is not differentiable at x_k, then $\tilde{s}_k \leftarrow -W_k G_k \omega_k = -W_k g_{k,1}$ might not be a descent direction for f from x_k. For this and other reasons (see [31, §4]), the line search might not be able to produce a stepsize within a prescribed iteration limit such that (4.1) holds. In such cases, one cannot guarantee the conditions of Theorem 3.4, namely, (3.10), since vanishing of $\{G_k \omega_k + \gamma_k\} = \{g_{k,1} + (\tilde{\alpha}_k - 1) g_{k,1}\}$ might not correspond to vanishing of (a subsequence of) $\{G_k \omega_k\} = \{g_{k,1}\}$ due to the large perturbations $\{\gamma_k\} = \{(\tilde{\alpha}_k - 1) g_{k,1}\}$. One can imagine various heuristics such that, if the line search would otherwise break down, one might replace $\tilde{s}_k$ with $-W_k G_k \omega_k$ for some matrix of subgradients G_k evaluated at points in $\mathbb{B}(x_k, \epsilon_k)$ for some $\epsilon_k \in \mathbb{R}_{++}$ and some nonnegative weight vector ω_k that sums to unity. However, it is a nontrivial task to determine such quantities—without a more sophisticated strategy such as that in §4.2—to ensure that the weak Wolfe line search will be guaranteed to return a stepsize above a prescribed positive threshold.

BFGS-Step

Require: A minimum stepsize parameter $\tilde{\alpha}_{\min} \in \mathbb{R}_{++}$.

1: Set $x_{k,1} \leftarrow x_k$ and $g_{k,1} \in \partial f(x_k)$.

2: Set $G_k \leftarrow [g_{k,1}]$, $\omega_k \leftarrow 1$, and $\tilde{s}_k \leftarrow -W_k G_k \omega_k$.

3: Run a weak Wolfe line search [31, Alg. 4.6] from x_k along $\tilde{s}_k$ to set $\tilde{\alpha}_k \in \mathbb{R}_+$ with

$$f(x_k + \tilde{\alpha}_k \tilde{s}_k) \leq f(x_k) + \tfrac{1}{2}\alpha \tilde{\alpha}_k^2 g_{k,1}^T \tilde{s}_k \quad \text{and} \quad \tilde{\alpha}_k \geq \tilde{\alpha}_{\min}, \tag{4.1}$$

or **terminate** (i.e., *break down*) if no such $\tilde{\alpha}_k$ is found within an iteration limit.

4: Set $s_k \leftarrow \tilde{\alpha}_k \tilde{s}_k$ (meaning $s_k \leftarrow -W_k(G_k\omega_k + \gamma_k)$ with $\gamma_k = (\tilde{\alpha}_k - 1)g_{k,1}$).

5: Set $x_{k+1} \leftarrow x_k + s_k$.

6: **return** (s_k, x_{k+1}) to Step 5 in SVANO.

All of this being said, *if* BFGS-Step yields $\tilde{\alpha}_k \geq \tilde{\alpha}_{\min}$ for all $k \in \mathbb{N}$, then the resulting instance of SVANO attains some of the guarantees in §3.2. In particular, with $\tilde{\alpha}_k \geq \tilde{\alpha}_{\min}$ for all $k \in \mathbb{N}$, the conditions of Theorem 3.4 hold, and if $g_{k,1} \in \partial f(x_k)$ or some heuristic is used to set $\tilde{s}_k \leftarrow -W_k G_k \omega_k$ as described in the previous paragraph, then the supposition about $\{g_{k,j}\}_{j=1}^m$ in Theorem 3.4 also holds. Consequently, *if* a subsequence of iterates converges to a limit and the subgradients employed in the step computation are evaluated at points in narrowing neighborhoods of each iterate, then a stationary point will be revealed by (a subsequence of) $\{x_k\}$.

Other classical BFGS variants of SVANO can be derived that employ a trust region mechanism instead of a line search. However, the issues for such a method would be similar to those described above: *if* one finds that a successful step is taken sufficiently often when the trust region radius is above a positive threshold (say, proportional to $\|G_k\omega_k\|_2$), then the algorithm has the potential to converge. However, it is nontrivial to design an algorithm that maintains the spirit of a basic quasi-Newton algorithm and ensures that this occurs under loose assumptions on f.

4.2 A bundle trust region method for convex minimization

Bundle methods are an extremely popular class of algorithms for solving nonsmooth optimization problems. Modern variants of bundle methods have guarantees for solving both convex [23, 26, 34, 41] and nonconvex [21, 22, 25, 28, 32, 33] problems. Those for solving convex problems are based on cutting plane and proximal point methodologies, whereas those for solving nonconvex problems employ the same ideas with "downshifting" and "tilting" of the cutting planes. We refer the reader to the references above for further information and details.

One can derive instances of SVANO using bundle method ideas that are intended for solving nonconvex problems. However, for simplicity, we present a particular instance designed for solving convex problems only; i.e., in this subsection, we make the following assumption.

Assumption 4.1. *The objective function f is convex.*

Our instance is described by specifying an algorithm for the step computation procedure written generically in Steps 2–4 in SVANO. In particular, see SVANO-Bundle-Step below. Through an inner loop, the procedure computes trial steps through successive subproblem solves until one is computed satisfying a descent condition, shown in our analysis below to imply that (2.7) holds. Each iteration of the inner loop involves solving a subproblem of the form (2.15) by solving the dual (2.17) for $(\omega_{k,m}, \gamma_{k,m}) \in \mathbb{R}^m_+ \times \mathbb{R}^n$ and recovering the primal solution as described in §2.3. (We now include a second subscript on the primal-dual subproblem solution vectors since they change as the algorithm iterates over $m \in \mathbb{N}$.) The "bundles" employed in the loop are the tuples $\{(x_{k,j}, f_{k,j}, g_{k,j})\}_{j=1}^m$ where $\{f_{k,j}\}_{j=1}^m$ and $\{g_{k,j}\}_{j=1}^m$ are objective values and subgradients evaluated at $\{x_{k,j}\}_{j=1}^m$. As in a standard bundle method, the bundles could also include elements of bundles computed in previous "outer" iterations, but, for simplicity and since it is not required for our convergence analysis, we do not state this option explicitly.

SVANO-Bundle-Step

1: Set $x_{k,1} \leftarrow x_k$, $f_{k,1} \leftarrow f(x_{k,1})$, and $g_{k,1} \in \partial f(x_{k,1})$.
2: **for** $m \in \mathbb{N}$ **do**
3: Set $G_{k,m} \leftarrow \begin{bmatrix} g_{k,1} & \cdots & g_{k,m} \end{bmatrix}$, then set $(\omega_{k,m}, \gamma_{k,m})$ by solving (2.17).
4: Set $x_{k,m+1} \leftarrow x_k - W_k(G_{k,m}\omega_{k,m} + \gamma_{k,m})$
5: Set $f_{k,m+1} \leftarrow f(x_{k,m+1})$ and $g_{k,m+1} \in \partial f(x_{k,m+1})$.
6: **if** $l_{k,m}(x_{k,m+1}) = f(x_k)$ **then terminate** since x_k is stationary for f.
7: **if** $f(x_k) - f_{k,m+1} \geq \alpha(f(x_k) - l_{k,m}(x_{k,m+1}))$ **then**
8: Set $s_k \leftarrow -W_k(G_{k,m}\omega_{k,m} + \gamma_{k,m})$ and $x_{k+1} \leftarrow x_{k,m+1}$.
9: **return** (s_k, x_{k+1}) to Step 5 in SVANO.
10: **end if**
11: **end for**

A convergence analysis for SVANO with step computations using SVANO-Bundle-Step is obtained by following a standard analysis for a bundle method (e.g., see [41, §7.4]), the main difference being that the analysis must pay careful attention to the properties of the (inverse) Hessian approximations. In this respect, the analysis relies on the self-correcting properties of BFGS-type updates discussed in §3.1. We present an analysis that proceeds in two stages: first, it is argued that, for any $k \in \mathbb{N}$ in which the iterate x_k is suboptimal, the inner loop will terminate finitely, ensuring that the algorithm is well-defined in the sense that it will either reach a minimizer of f in a finite number of iterations or generate an infinite sequence of outer iterates; second, it is argued that each accepted step yields a sufficient reduction in f such that at least one limit point of the outer iteration sequence is a solution of problem (P). Since some of our results follow standard techniques in analyses of bundle methods (e.g, see [41, §7.4]), some lemmas and theorems are stated here without proofs; complete proofs can be found in Appendix 4.2.

Toward proving that SVANO-Bundle-Step is well-defined, one may use a type of *Moreau-Yosida regularization* function of f corresponding, for a given $k \in \mathbb{N}$, to the symmetric positive-definite matrix H_k and trust region $\mathcal{X}_k$; specifically, consider the function $f_{H_k,\mathcal{X}_k} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f_{H_k,\mathcal{X}_k}(\overline{x}) = \min_{x \in \mathcal{X}_k} f(x) + \tfrac{1}{2}(x - \overline{x})^T H_k (x - \overline{x}). \tag{4.2}$$

This function provides a mechanism for quantifying the separation between f and the model functions $l_{k,m}$ and $q_{k,m}$ defined in (2.13) and (2.14), respectively. To start, the following lemma states that if x_k is not a minimizer of f, then this Moreau-Yosida regularization function's value at x_k is strictly less than the objective function value at x_k. The proof of the result is based simply on the existence of another point in $\mathbb{R}^n$ that yields a better objective function value than does x_k.

Lemma 4.2. *For any $k \in \mathbb{N}$, if x_k does not minimize f, then $f_{H_k,\mathcal{X}_k}(x_k) < f(x_k)$.*

On the other hand, the next lemma shows that the Moreau-Yosida regularization function offers an upper bound for the piecewise-linear and piecewise-quadratic model values corresponding to the optimal solution of (2.15).

Lemma 4.3. *For any $(k, m) \in \mathbb{N} \times \mathbb{N}$, the value of $l_{k,m}$ evaluated at $x_{k,m+1}$ is bounded above by the optimal value of (2.15), which, in turn, is bounded above by the Moreau-Yosida regularization function $f_{H_k,\mathcal{X}_k}$ evaluated at x_k; i.e.,*

$$l_{k,m}(x_{k,m+1}) \leq q_{k,m}(x_{k,m+1}) \leq f_{H_k,\mathcal{X}_k}(x_k). \tag{4.3}$$

The following lemma now shows that the inner loop of the algorithm is well-defined in the sense that if a minimizer of f has not yet been obtained, then the algorithm will eventually compute a point satisfying the condition in Step 7. The proof follows analyses as found in [41, Chap. 7], but we provide it here for completeness.

Lemma 4.4. *For any $k \in \mathbb{N}$, if x_k is not a minimizer of f and $\delta_k \in (0,\infty)$, then* SVANO-Bundle-Step *terminates.*

Proof. Let $k \in \mathbb{N}$ be given and suppose that x_k is not a minimizer of f. Then, for any $m \in \mathbb{N}$, Lemmas 4.2 and 4.3 imply that

$$l_{k,m}(x_{k,m+1}) \leq q_{k,m}(x_{k,m+1}) \leq f_{H_k,\mathcal{X}_k}(x_k) < f(x_k), \tag{4.4}$$

meaning that the termination check in Step 6 never tests true. Hence, to derive a contradiction to the statement of the lemma, suppose that the algorithm generates an infinite sequence $\{x_{k,m+1}\}_{m=1}^{\infty}$ such that no element satisfies the condition in Step 7.

Toward deriving the aforementioned contradiction, let us first show that the generated function values $\{f(x_{k,m+1})\}_{m=1}^{\infty}$ converge to the minimizer of f over $\mathcal{X}_k$, namely, $f_{\mathcal{X}_k} := \min_{x\in\mathcal{X}_k} f(x)$. Notice that since x_k is not a minimizer of f, it follows by convexity of f that $f_{\mathcal{X}_k} < f(x_k)$. For any $\varepsilon \in (0,\infty)$, let

$$\mathcal{M}_\varepsilon := \{m \in \mathbb{N} : f_{\mathcal{X}_k} + \varepsilon < f(x_{k,m+1})\}.$$

Suppose that there exist two indices in this set, say $(m_1, m_2) \in \mathcal{M}_\varepsilon \times \mathcal{M}_\varepsilon$ with $m_1 < m_2$. Since $m_1 < m_2$ and $\{l_{k,m}\}_{m=1}^{\infty}$ are pointwise underestimators of f, we can conclude that

$$f_{k,m_1} + g_{k,m_1}^T(x_{k,m_2+1} - x_{k,m_1}) \leq l_{k,m_2}(x_{k,m_2+1}) \leq f_{\mathcal{X}_k}. \tag{4.5}$$

On the other hand, by virtue of m_2 being an element of $\mathcal{M}_\varepsilon$, it follows that $\varepsilon < f(x_{k,m_2+1}) - f_{\mathcal{X}_k}$, which combined with (4.5) implies that

$$\varepsilon < f(x_{k,m_2+1}) - f_{k,m_1} - g_{k,m_1}^T(x_{k,m_2+1} - x_{k,m_1}). \tag{4.6}$$

Since $\mathcal{X}_k$ is compact, there exists $L_{\mathcal{X}_k} \in (0,\infty)$ such that (recall (2.2))

$$|f(x) - f(\bar{x})| \leq L_{\mathcal{X}_k}\|x - \bar{x}\|_2 \quad \text{for all} \quad (x,\bar{x}) \in \mathcal{X}_k \times \mathcal{X}_k.$$

Since the subgradients of f on $\mathcal{X}_k$ are bounded, one can assume that $L_{\mathcal{X}_k}$ is large enough such that $\|g_{k,m+1}\|_2 \leq L_{\mathcal{X}_k}$ for all $m \in \mathbb{N}$. Hence, from (4.6), one finds

$$\varepsilon < 2L_{\mathcal{X}_k}\|x_{k,m_2+1} - x_{k,m_1}\|_2 \quad \text{for all} \quad (m_1,m_2) \in \mathcal{M}_\varepsilon \times \mathcal{M}_\varepsilon \quad \text{with} \quad m_1 \neq m_2.$$

Since the set $\mathcal{X}_k$ is compact, there can only exist a finite number of points in $\mathcal{X}_k$ having a distance at least $\varepsilon/(2L_{\mathcal{X}_k})$ from each other. Thus, the set $\mathcal{M}_\varepsilon$ must be finite. In turn, this means that for any $\varepsilon \in (0,\infty)$ there can only be a finite number of points with objective function value in $[f_{\mathcal{X}_k}, f_{\mathcal{X}_k} + \varepsilon]$. One may thus conclude that the sequence $\{f(x_{k,m+1})\}_{m=1}^{\infty}$ converges to $f_{\mathcal{X}_k}$.

Let us now use the established convergence of $\{f(x_{k,m+1})\}_{m=1}^{\infty}$ to $f_{\mathcal{X}_k}$ to derive a contradiction to the supposition that $\{x_{k,m+1}\}_{m=1}^{\infty}$ is generated with no element satisfying the condition in Step 7. Since $\{x_{k,m+1}\}_{m=1}^{\infty}$ is contained in the compact set $\mathcal{X}_k$, there exists an infinite $\mathcal{M} \subseteq \mathbb{N}$ such that

$$\lim_{m\in\mathcal{M},m\to\infty} x_{k,m+1} = \bar{x} \quad \text{for some} \quad \bar{x} \in \mathcal{X}_k.$$

Since $\{f(x_{k,m+1})\}_{m=1}^{\infty} \to f_{\mathcal{X}_k}$, it follows that $f(\bar{x}) = f_{\mathcal{X}_k}$. For any $m \in \mathcal{M}$, let $\overline{m}$ be the smallest element in $\mathcal{M}$ that is strictly larger than m. It follows using the same argument that lead to (4.5) that

$$f(x_{k,m+1}) + g_{k,m+1}^T(x_{k,\overline{m}+1} - x_{k,m+1}) \leq l_{l,\overline{m}}(x_{k,\overline{m}+1}) \leq f_{\mathcal{X}_k}.$$

Taking limits over $m \in \mathcal{M}$ as $m \to \infty$, using the uniform bound on the subgradient norms $\{\|g_{k,m+1}\|\}_{m=1}^{\infty}$ over $\mathcal{X}_k$ as enlisted in the previous paragraph, and recalling the facts that $\lim_{m\in\mathcal{M},m\to\infty} x_{k,m+1} = \bar{x}$ and $f(\bar{x}) = f_{\mathcal{X}_k}$, one finds that

$$\begin{aligned} f_{\mathcal{X}_k} = f(\bar{x}) &= \lim_{m\in\mathcal{M},m\to\infty} \left(f(x_{k,m+1}) + g_{k,m+1}^T(x_{k,\overline{m}+1} - x_{k,m+1})\right) \\ &\leq \lim_{m\in\mathcal{M},m\to\infty} l_{k,\overline{m}}(x_{k,\overline{m}+1}) \leq f_{\mathcal{X}_k}. \end{aligned}$$

Since $\lim_{m\in\mathcal{M},m\to\infty} l_{k,m}(x_{k,m+1}) = \lim_{m\in\mathcal{M},m\to\infty} l_{k,\overline{m}}(x_{k,\overline{m}+1})$, this proves that

$$\lim_{m\in\mathcal{M},m\to\infty} l_{k,m}(x_{k,m+1}) = f_{\mathcal{X}_k},$$

from which it follows that

$$\lim_{m\in\mathcal{M},m\to\infty} \left(\frac{f(x_k) - f(x_{k,m+1})}{f(x_k) - l_{k,m}(x_{k,m+1})}\right) = \frac{f(x_k) - f_{\mathcal{X}_k}}{f(x_k) - f_{\mathcal{X}_k}} = 1.$$

We have reached a contradiction since this limit indicates that the condition in Step 7 would be satisfied for some sufficiently large $m \in \mathcal{M}$. □

We have thus shown that, unless the algorithm lands on a point that is stationary for f, the step computation procedure terminates for any $k \in \mathbb{N}$. Since the condition in Step 7 implies (2.20), it follows from Lemma 2.4 that the computed step satisfies (2.7), meaning that the overall algorithm is well posed. Thus, all that remains is to show that this instance of SVANO satisfies the remaining assumptions of Theorem 3.4. This is done in the following theorem when one introduces a particular strategy for updating the trust region radius during the algorithm.

Theorem 4.5. *Suppose that one chooses $\delta_1 \in (0,\infty)$ and a parameter $\tau \in (0,1)$. In addition, suppose that, at the end of each iteration $k \in \mathbb{N}$ in* SVANO*, one sets*

$$\delta_{k+1} \leftarrow \begin{cases} \tau\delta_k & \text{if } \max\{\upsilon_1\|G_k\omega_k + \gamma_k\|_2, \upsilon_2\|s_k\|_2, \upsilon_3\|G_k\omega_k\|_2\} \leq \delta_k \\ \delta_k & \text{otherwise.} \end{cases} \tag{4.7}$$

for some prescribed constants $(\upsilon_1,\upsilon_2,\upsilon_3) \in \mathbb{R}_{++}\times\mathbb{R}_{++}\times\mathbb{R}_{++}$. Then, there exists an infinite index set $\mathcal{K}' \subseteq \mathbb{N}$ such that

$$\lim_{k\in\mathcal{K}',k\to\infty} \|G_k\omega_k\|_2 = 0 \tag{4.8}$$

with any limit point of $\{x_k\}_{k\in\mathcal{K}'}$ being stationary for f.

Proof. Our first goal is to show that, with the update (4.7), one finds $\{\delta_k\} \searrow 0$. To derive a contradiction, suppose that there exists $\delta \in (0,\delta_1]$ such that $\delta_k = \delta$ for all sufficient large $k \in \mathbb{N}$. For a given $p \in (0,1)$, let $\mathcal{K} \subseteq \mathbb{N}$ be the infinite index set for which Theorem 3.1 and Corollary 3.2 guarantee the existence of $(\kappa,\sigma,\mu) \in \mathbb{R}_{++}\times\mathbb{R}_{++}\times\mathbb{R}_{++}$ and $(\nu,\xi) \in \mathbb{R}_{++}\times\mathbb{R}_{++}$ such that (3.5) and (3.6) hold, and, by Theorem 3.3, such that (3.8) holds. It follows that, for all sufficiently large $k \in \mathcal{K}$, the strict inequality $\upsilon_3\|G_k\omega_k\|_2 > \delta$ must hold, or else the update (4.7) would eventually reduce the trust region radius below δ. Since this and the triangle inequality yield

$$\begin{aligned} \|G_k\omega_k + \gamma_k\|_2 \geq \|G_k\omega_k\|_2 - \|\gamma_k\|_2 > \delta/\upsilon_3 - \|\gamma_k\|_2 \\ \implies \|\gamma_k\|_2 > \delta/\upsilon_3 - \|G_k\omega_k + \gamma_k\|_2 \end{aligned}$$

for all sufficiently large $k \in \mathcal{K}$, it follows along with the first limit in (3.8) that

$$\liminf_{k\in\mathcal{K},k\to\infty} \|\gamma_k\|_2 \geq \delta/\upsilon_3.$$

This implies that the trust region radius is active for all sufficiently large $k \in \mathcal{K}$, which contradicts the second limit in (3.8). As a result, we must conclude that there exists an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ such that (4.7) yields $\delta_{k+1} \leftarrow \tau\delta_k$ for all $k \in \mathcal{K}'$, which by (4.7) implies that (4.8) holds. This limit, along with the fact that the subgradients used in SVANO-Bundle-Step are evaluated at points in $\mathbb{B}(x_k,\delta_k)$ for all $k \in \mathbb{N}$, implies by Theorem 3.4 that any limit point of $\{x_k\}_{k\in\mathcal{K}'}$ is stationary for f. □

5 Numerical Experiments

We implemented SVANO in C++. The implementation includes input options that offer the following algorithms.

- `BFGS`: This follows the strategy described in §4.1 where steps are computed using a standard BFGS strategy and a weak Wolfe line search is employed. (We do not impose the trust region constraint for the subproblems for this algorithm, though we do generate the sequence $\{\delta_k\}$ for use in our termination conditions; see below.) As previously mentioned, we do not claim a convergence guarantee for this method.

- `SVANO-Bundle`: This follows the strategy described in §4.2 where steps are computed using a bundle trust region strategy. However, for when a problem is nonconvex, we replace $b_{k,j}$ in (2.18) with

$$b_{k,j} = \min\{f(x_k), f_{k,j} + g_{k,j}^T(x_k - x_{k,j})\}, \tag{5.1}$$

  which, if the first term in the minimum is smaller, causes a "downshifting" of a cutting plane for the subproblem. In addition, after a step is computed satisfying (2.7), a weak Wolfe line search is employed (with an upper bound for the stepsize imposed so that the norm of the resulting step is proportional to the trust region radius). This significantly improved the performance of the algorithm, we believe by yielding better steps and better pairs for the BFGS updating strategy. We claim that this additional procedure does not ruin the convergence guarantees stated in §4.2 when solving convex problems.

- `SVANO-GS`: This follows an adaptive gradient sampling strategy along the lines of that proposed in [11, 12] with the addition of a trust region constraint (with radius δ_k for all $k \in \mathbb{N}$). Convergence guarantees for this algorithm can be ensured by combining the analysis in this paper with that in [12].

Our implementation of SVANO allows for various choices of $\overline{H}$ and y_k for all $k \in \mathbb{N}$. For example, as often proposed for BFGS methods for smooth optimization (e.g., see [35]), one might choose the former as a multiple of the identity where the multiplying factor is determined by a Barzilai-Borwein "two-point stepsize" strategy [1], projected onto $[\eta, \theta]$, after the first accepted step. One could even update it with each iteration, as long as the factor is projected onto $[\eta, \theta]$ for all $k \in \mathbb{N}$. Another strategy would be to initialize $H_1 \leftarrow I$, employ (2.11) through iteration $K \in \mathbb{N}$, then set $\overline{H} \leftarrow H_{K+1}$ for use in all subsequent iterations. As for y_k, one could choose the gradient displacement $\nabla f(x_{k+1}) - \nabla f(x_k)$ as in a standard BFGS strategy, or choose $\nabla f(x_{k+1}) - G_k\omega_k$, where the latter term is a convex combination of gradients computed at points about x_k. However, while our algorithmic framework and software allow these various choices, for our experiments here we simply set $W_1 \leftarrow I$ and $y_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$ for all $k \in \mathbb{N}$ so that our comparison of the algorithms mentioned above would be based on common choices of these values.

Our code updates sequences the $\{\epsilon_k\}$ and $\{\delta_k\}$ dynamically, the former playing a role in our termination conditions (see below) and as the sampling radii for `SVANO-GS`, and the latter playing the role of trust region radii for `SVANO-Bundle` and `SVANO-GS`. In addition, at the beginning of each iteration $k \in \mathbb{N}$ of `SVANO-Bundle` and `SVANO-GS`, the code initializes the set $\{x_{k+1,j}\}$ with those points from $\{x_{k,j}\}$ that are within $\mathbb{B}(x_{k+1}, \delta_{k+1})$. The sequence $\{\delta_k\}$ is updated dynamically according to (4.7) with $\delta_1 \leftarrow 10$ and $\epsilon_k \equiv 10^{-2}\delta_k$ for all $k \in \mathbb{N}$. The remaining inputs used in the code were $\alpha \leftarrow 10^{-8}$, $\eta \leftarrow 10^{-12}$, $\theta \leftarrow 10$, and, in the update (4.7), $\upsilon_1 = \upsilon_2 = \upsilon_3 \leftarrow 1$. The parameter θ we found to have a particularly large impact on performance; see later on in this section for further discussion.

For test problems, we used the first ten from [20] with $n = 50$. Pertinent information about the problems for this dimension n—namely, an indication of whether each problem is convex, the objective value at the initial point ($f(x_0)$), and the global minimum value of the objective ($f(x_*)$)—are given in Table 1. For further information, including the starting point for each problem; see [20].

Consistent with our theoretical analysis, the code terminates with a message of success when it observes, for some $k \in \mathbb{N}$, that

$$\|G_k\omega_k\|_2 \leq 10\epsilon_k \quad \text{and} \quad \epsilon_k \leq 10^{-4}. \tag{5.2}$$

Name	Convex?	$f(x_0)$	$f(x_*)$
maxq	Yes	2500.0	0.0
mxhilb	Yes	4.5	0.0
chained lq	Yes	49.0	69.3
chained cb3 1	Yes	980.0	98.0
chained cb3 2	Yes	980.0	98.0
active faces	No	3.9	0.0
brown function 2	No	98.0	0.0
chained mifflin 2	No	232.8	-34.8
chained crescent 1	No	292.3	0.0
chained crescent 2	No	292.3	0.0

Table 1: Test problem information.

The code terminates with a message of failure if the iteration limit of 10^4 is reached or a computed stepsize is below 10^{-20}, the latter playing the role of $\tilde{\alpha}_{\min}$ in Algorithm BFGS-Step. The results obtained with these stopping conditions are shown in Table 2. In the table, an `Exit` of `Stationary` indicates a successful termination while that of `Iteration` or `Stepsize` indicates that the iteration or stepsize limit, respectively, was exceeded. The values $\epsilon_{\texttt{eps}}$ and $f(x_{\texttt{eps}})$ indicate the final elements of the sequences $\{\epsilon_k\}$ and $\{f(x_k)\}$ at termination, and the counters `#iter`, `#func`, `#grad`, and `#subs` indicate the number of iterations, function evaluations, gradient evaluations, and subproblems solved prior to termination.

One can see in these results that `SVANO-Bundle` and `SVANO-GS` behave quite well in the sense that they terminate with success for all problems. Which algorithm performs the best for a particular problem depends on the performance measure of interest. In particular, `SVANO-Bundle` often requires fewer function evaluations, but often at the expense of more gradient evaluations and subproblem solves.

More striking in the results in Table 2 is the fact that `BFGS` only terminated with a message of success for one problem. For the remaining problems, the code terminated due to a small stepsize (below 10^{-20}). This provides evidence for our discussion in §4.1, where we stated that the main issue with proving convergence guarantees for a classical BFGS approach is that one cannot be sure that the stepsize would remain sufficiently large. That being said, the final objective values yielded by `BFGS` show that this code did not always perform poorly in terms of the final objective value! For many problems, the final value was close to optimal.

To illustrate the benefits of self-correction, we also ran the experiments with the same settings, except with $\theta \leftarrow \infty$, where in particular the latter choice does *not* satisfy our restrictions to ensure our convergence guarantees. The results with these inputs are given in Table 3. Clearly, the performance is not as good. In particular, the final objective values are often nearly optimal, but the code has a difficult time satisfying our termination criteria. Indeed, this has been observed before for BFGS-type methods for nonsmooth optimization, where the proposed remedy is either to switch to a convergent algorithm after BFGS "breaks down". Overall, a benefit of our self-correcting framework is that this second phase is not needed.

6 Conclusion

We have proposed an algorithmic framework for solving nonsmooth, nonconvex optimization problems. The distinguishing characteristic of the framework is that it maintains and benefits from the self-correcting properties of BFGS updating of the generated sequence of inverse Hessian approximations. In particular, it benefits theoretically in that global convergence guarantees can be established, and it benefits in practice in that instances of the framework are effectively able to determine when iterates are nearly stationary for the objective function.

It is worthwhile to point out that we have not discussed limited-memory BFGS, even though using limited memory ideas is another alternative for ensuring that the inverse Hessian approximations have eigenvalues that are uniformly bounded below (away from zero) and above. The primary reason for this omission is that

BFGS							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+2.26e-07	450	1017	452	451
mxhilb	Stepsize	+3.13e-03	+9.26e-02	96	1658	107	97
chained lq	Stepsize	+5.00e-02	-6.93e+01	196	4345	198	197
chained cb3 1	Stepsize	+1.00e-01	+9.80e+01	399	8925	400	400
chained cb3 2	Stepsize	+1.00e-01	+9.80e+01	82	2192	91	83
active faces	Stepsize	+2.50e-02	+2.22e-16	24	672	27	25
brown function 2	Stepsize	+1.00e-01	+1.95e-05	2130	119179	2131	2131
chained mifflin 2	Stepsize	+5.00e-02	-3.47e+01	453	8940	496	454
chained crescent 1	Stepsize	+1.00e-01	+2.18e-01	74	2278	91	75
chained crescent 2	Stepsize	+1.00e-01	+5.87e-02	316	8592	320	317
SVANO-Bundle							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+1.04e-06	193	441	635	440
mxhilb	Stationary	+9.77e-05	+4.09e-05	47	271	278	150
chained lq	Stationary	+9.77e-05	-6.93e+01	29	374	398	366
chained cb3 1	Stationary	+9.77e-05	+9.80e+01	65	2024	1996	1929
chained cb3 2	Stationary	+9.77e-05	+9.80e+01	27	155	183	154
active faces	Stationary	+9.77e-05	+2.09e-02	17	440	218	31
brown function 2	Stationary	+9.77e-05	+4.34e-03	480	23574	22738	22252
chained mifflin 2	Stationary	+9.77e-05	-3.48e+01	246	14274	11936	11578
chained crescent 1	Stationary	+9.77e-05	+2.73e-04	30	66	92	59
chained crescent 2	Stationary	+9.77e-05	+4.36e-05	137	6679	6140	5997
SVANO-GS							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+1.28e-06	261	520	268	262
mxhilb	Stationary	+9.77e-05	+1.26e-04	56	768	321	74
chained lq	Stationary	+9.77e-05	-6.93e+01	3084	39106	3450	3125
chained cb3 1	Stationary	+9.77e-05	+9.80e+01	2585	34137	2912	2666
chained cb3 2	Stationary	+9.77e-05	+9.80e+01	73	485	90	84
active faces	Stationary	+9.77e-05	+4.09e-03	15	458	232	16
brown function 2	Stationary	+9.77e-05	+1.54e-02	149	3005	333	302
chained mifflin 2	Stationary	+9.77e-05	-3.48e+01	2273	35719	3124	2487
chained crescent 1	Stationary	+9.77e-05	+1.91e-04	22	144	33	27
chained crescent 2	Stationary	+9.77e-05	+3.78e-03	1938	27472	2391	2055

Table 2: Termination status, solution properties, and counter values when `BFGS`, `SVANO-Bundle`, and `SVANO-GS` were employed to solve the test problems stated in Table 1.

it has been widely observed that limited memory BFGS techniques do not always perform as well as a full memory approach in the context of nonsmooth optimization.

7 Acknowledgement

The authors would like to thank Andreas Wächter for hosting and providing guidance to the third author during that author's visit to Northwestern University in the summer of 2017 while that author was implementing updates to the quadratic optimization solver for our software package.

A Geometric Properties of BFGS Updating

The update (2.11) performs a *projection* to eliminate certain curvature dictated by the Hessian approximation H_k, as well as a corresponding *correction* that replaces this curvature for the new approximation H_{k+1}. The details of this projection and associated correction in the Hessian approximation can be seen in the following manner.[1] For the sake of generality, let $H \in \mathbb{R}^{n\times n}$ be symmetric positive definite, i.e., $H \succ 0$. An inner product based on H (i.e., an "H-inner product") is

$$\langle s, v\rangle_H := s^T H v \quad \text{for all} \quad (s, v) \in \mathbb{R}^n \times \mathbb{R}^n.$$

[1]The presentation in this appendix is based on notes by James V. Burke, Adrian S. Lewis, and Michael O. Overton, which were shared with the first author by Michael O. Overton.

BFGS (not enforcing (2.9))							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+2.37e-07	543	1478	643	544
mxhilb	Stepsize	+1.56e-03	+3.83e-09	966	10632	1070	967
chained lq	Stepsize	+5.00e-02	-6.93e+01	341	2355	343	342
chained cb3 1	Stepsize	+1.00e-01	+9.80e+01	506	2799	507	507
chained cb3 2	Iteration	+1.00e-01	+9.80e+01	10000	324980	10002	10001
active faces	Stepsize	+2.50e-02	+4.44e-15	39	322	42	40
brown function 2	Stepsize	+1.00e-01	+2.59e-11	272	8751	293	273
chained mifflin 2	Stepsize	+5.00e-02	-3.48e+01	1597	4143	1774	1598
chained crescent 1	Iteration	+1.00e-01	+3.17e-12	10000	507638	10016	10001
chained crescent 2	Stepsize	+1.00e-01	+1.93e-12	538	3070	543	539
SVANO-Bundle (not enforcing (2.9))							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+1.21e-06	135	367	503	366
mxhilb	Stepsize	+1.95e-04	+1.47e-07	94	1155	818	704
chained lq	Stepsize	+1.25e-02	-6.93e+01	132	2199	2229	2095
chained cb3 1	Stepsize	+2.50e-02	+9.80e+01	240	4796	4912	4670
chained cb3 2	Iteration	+6.25e-03	+9.80e+01	10000	907606	508821	498819
active faces	Stationary	+9.77e-05	+1.92e-06	19	504	217	30
brown function 2	Stepsize	+6.25e-03	+3.69e-04	429	6483	6560	6123
chained mifflin 2	Stepsize	+2.50e-02	-3.48e+01	194	6849	4517	4317
chained crescent 1	Stationary	+9.77e-05	+7.27e-04	29	55	83	51
chained crescent 2	Stepsize	+1.25e-02	+1.01e-02	164	5081	5138	4973
SVANO-GS (not enforcing (2.9))							
Name	Exit	ϵ_{end}	$f(x_{\text{end}})$	#iter	#func	#grad	#subs
maxq	Stationary	+9.77e-05	+1.13e-06	236	799	305	237
mxhilb	Stationary	+9.77e-05	+1.03e-06	91	541	342	92
chained lq	Stepsize	+6.25e-03	-6.93e+01	691	4441	1409	727
chained cb3 1	Stepsize	+1.25e-02	+9.86e+01	1037	5638	2029	1047
chained cb3 2	Iteration	+7.81e-04	+9.80e+01	10000	443532	10013	10001
active faces	Stationary	+9.77e-05	+2.42e-03	14	576	285	16
brown function 2	Stationary	+9.77e-05	+6.02e-03	147	1915	498	197
chained mifflin 2	Stepsize	+6.25e-03	-3.48e+01	1581	10350	4576	1621
chained crescent 1	Stationary	+9.77e-05	+9.05e-06	65	174	77	67
chained crescent 2	Stepsize	+2.50e-02	+6.58e-01	304	1917	376	323

Table 3: Termination status, solution properties, and counter values when `BFGS`, `SVANO-Bundle`, and `SVANO-GS` were employed to solve the test problems stated in Table 1. In contrast to the results shown in Table 2, these results were obtained with $\theta \leftarrow \infty$, meaning that the latter bound in (2.9) is not enforced, which nullifies our convergence guarantees. This has an adverse effect on performance.

Given a matrix $A \in \mathbb{R}^{n\times n}$, let its "$H$-adjoint" with respect to this H-inner product, call it $A^* \in \mathbb{R}^{n\times n}$, be a matrix that satisfies

$$\langle s, Av\rangle_H = \langle A^* s, v\rangle_H \quad \text{for all} \quad (s,v) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Since $H \succ 0$, it is easily verified that the unique H-adjoint of A is $A^* = H^{-1}A^T H$. One calls A an "H-orthogonal" projection matrix if and only if it satisfies

$$\begin{aligned} A &= A^2 \quad \text{(i.e., } A \text{ is idempotent)} \\ \text{and} \quad A &= A^* \quad \text{(i.e., } A \text{ is "}H\text{-self-adjoint").} \end{aligned}$$

For example, given a nonzero vector $s \in \mathbb{R}^n$, consider the matrices

$$P := \frac{ss^T H}{s^T H s} \quad \text{and} \quad Q := I - P.$$

It happens that P and Q are both H-orthogonal projection matrices. In particular, P yields the H-orthogonal projection onto $\text{span}(s)$ while Q yields the H-orthogonal projection onto the subspace H-orthogonal to $\text{span}(s)$. That is, given $t \in \mathbb{R}^n$ such that $\langle s, t\rangle_H = 0$ (i.e., t lies in the subspace H-orthogonal to $\text{span}(s)$), it follows that

$$\left\{\begin{matrix} Ps = s \\ Qs = 0 \end{matrix}\right\} \quad \text{while} \quad \left\{\begin{matrix} Pt = 0 \\ Qt = t \end{matrix}\right\}.$$

One may now interpret the updates yielded by (2.11) in terms of sequences of projections and corrections. Specifically, note that (2.11) can be rewritten as

$$H_{k+1} \leftarrow H_k^{(n-1)} + H_k^{(1)} \tag{A.1a}$$

$$\text{where} \quad H_k^{(n-1)} := Q_k^T H_k Q_k \quad \text{with} \quad Q_k := \left(I - \frac{s_k s_k^T H_k}{s_k^T H_k s_k}\right) \tag{A.1b}$$

$$\text{and} \quad H_k^{(1)} := \frac{v_k v_k^T}{s_k^T v_k}. \tag{A.1c}$$

Based on the discussion above, Q_k yields the H_k-orthogonal projection onto the subspace H_k-orthogonal to $\text{span}(s_k)$. Looking more closely, $H_k^{(n-1)}$ has rank $n-1$, remains positive definite on the subspace H-orthogonal to $\text{span}(s_k)$, and

$$H_k^{(n-1)} s_k = 0 \quad \text{while} \quad H_k^{(n-1)} t = H_k t \quad \text{if} \quad \langle s_k, t \rangle_H = 0. \tag{A.2}$$

On the other hand, the matrix $H_k^{(1)}$ can be written as

$$H_k^{(1)} = \frac{v_k v_k^T}{s_k^T v_k} = \left(\frac{\|v_k\|_2^2}{s_k^T v_k}\right)\left(\frac{v_k v_k^T}{\|v_k\|_2^2}\right),$$

where, in light of the secant-like equation (3.1), the leading scalar $\|v_k\|_2^2 / s_k^T v_k = \|v_k\|_2^2 / v_k^T W_{k+1} v_k$ is the inverse of a Rayleigh quotient for W_{k+1} defined by v_k. Since the so-called "curvature condition" $s_k^T v_k > 0$ holds by construction in (2.9), one finds that $s_k^T H_k^{(1)} s_k = s_k^T v_k > 0$, so, observing (A.1a), one finds that $H_k^{(1)}$ corrects the curvature along $\text{span}(s_k)$ that, according to (A.2), has been projected out of $H_k^{(n-1)}$.

B Primal and Dual Subproblems

In this appendix, we show that the dual of problem (2.15) is given by problem (2.17), how the solution of (2.15) can be recovered by that of (2.17), and that the remaining parts of Lemma 2.4 hold true.

As previously mentioned in §2.3, the primal problem (2.15) is equivalent to (2.16). A Lagrangian for this problem, call it $L : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$, is given by

$$L(x, z, \omega) = z + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) + \sum_{j=1}^{m} \omega_j (f_{k,j} + g_{k,j}^T (x - x_{k,j}) - z),$$

with which we can write the dual problem [4] for (2.16) as

$$\sup_{\omega \in \mathbb{R}_+^m} \inf_{(x,z) \in \mathcal{X}_k \times \mathbb{R}} L(x, z, \omega).$$

Differentiating the Lagrangian with respect to z, one finds that the "inner" infimum problem attains its minimum only if $\mathbb{1}^T \omega = 1$, from which it follows that the dual problem is equivalent to the constrained problem

$$\begin{aligned} &\sup_{\omega \in \mathbb{R}_+^m} \left(\inf_{x \in \mathcal{X}_k} \left(\tfrac{1}{2}(x - x_k)^T H_k (x - x_k) + \sum_{j=1}^{m} \omega_j (f_{k,j} + g_{k,j}^T (x - x_{k,j})) \right) \right) \\ &\quad \text{s.t. } \mathbb{1}^T \omega = 1. \end{aligned} \tag{B.1}$$

Defining the characteristic $\chi_{\mathcal{X}_k} : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ as one that evaluates as 0 for $x \in \mathcal{X}_k$ and ∞ otherwise, the inner infimum problem can equivalently be written as

$$\inf_{x \in \mathbb{R}^n} \overline{L}(x) + \chi_{\mathcal{X}_k}(x), \tag{B.2}$$

where we define the quadratic function $\overline{L} : \mathbb{R}^n \to \mathbb{R}$ by

$$\begin{aligned}\overline{L}(x) &= \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) + \sum_{j=1}^m \omega_j (f_{k,j} + g_{k,j}^T (x - x_{k,j})) \\ &= \tfrac{1}{2} x^T H_k x + x^T \left(-H_k x_k + \sum_{j=1}^m \omega_j g_{k,j} \right) + \tfrac{1}{2} x_k^T H_k x_k + \sum_{j=1}^m \omega_j (f_{k,j} - g_{k,j}^T x_{k,j}).\end{aligned}$$

The conjugate of $\overline{L}$, namely $\overline{L}^\star : \mathbb{R}^n \to \mathbb{R}$, is given by[2]

$$\begin{aligned}\overline{L}^\star(y) = \tfrac{1}{2} &\left(y + \left(H_k x_k - \sum_{j=1}^m \omega_j g_{k,j} \right) \right)^T W_k \left(y + \left(H_k x_k - \sum_{j=1}^m \omega_j g_{k,j} \right) \right) \\ &- \tfrac{1}{2} x_k^T H_k x_k - \sum_{j=1}^m \omega_j (f_{k,j} - g_{k,j}^T x_{k,j}) \\ = \tfrac{1}{2} &\left(y - \sum_{j=1}^m \omega_j g_{k,j} \right)^T W_k \left(y - \sum_{j=1}^m \omega_j g_{k,j} \right) \\ &+ x_k^T y - \sum_{j=1}^m \omega_j (f_{k,j} + g_{k,j}^T (x_k - x_{k,j})).\end{aligned}$$

In addition, the conjugate of $\chi_{\mathcal{X}_k}$, namely $(\chi_{\mathcal{X}_k})^\star : \mathbb{R}^n \to \mathbb{R}$, is given by

$$(\chi_{\mathcal{X}_k})^\star(y) = \sup_{x \in \mathbb{R}^n} (y^T x - \chi_{\mathcal{X}_k}(x)) = \sup_{x \in \mathcal{X}_k} y^T x = \sup_{\|x - x_k\| \le \delta_k} y^T x.$$

If $\delta_k = \infty$, then $(\chi_{\mathcal{X}_k})^\star(y) = \infty$ for all nonzero $y \in \mathbb{R}^n$. Otherwise, defining the vector $s := (x - x_k)/\delta_k$ so that $x = \delta_k s + x_k$, the above implies that

$$(\chi_{\mathcal{X}_k})^\star(y) = \sup_{\|x - x_k\| \le \delta_k} y^T x = \sup_{\|s\| \le 1} y^T (\delta_k s + x_k) = x_k^T y + \delta_k \|y\|_\star.$$

In either case, since the intersection of the relative interiors of the effective domains of $\overline{L}$ and χ_k is nonempty, Fenchel duality implies the strong duality relationship

$$\begin{aligned}\inf_{x \in \mathbb{R}^n} \ \overline{L}(x) + \chi_{\mathcal{X}_k}(x) &= \sup_{y \in \mathbb{R}^n} \ -\overline{L}^\star(y) - (\chi_{\mathcal{X}_k})^\star(-y) \\ &= \sup_{y \in \mathbb{R}^n} \ -\tfrac{1}{2} \left(y - \sum_{j=1}^m \omega_j g_{k,j} \right)^T W_k \left(y - \sum_{j=1}^m \omega_j g_{k,j} \right) \\ &\qquad + \sum_{j=1}^m \omega_j (f_{k,j} + g_{k,j}^T (x_k - x_{k,j})) - \delta_k \|y\|_\star.\end{aligned}$$

[2]Recall that for $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$ with $A \succ 0$, the conjugate of $\overline{L} : \mathbb{R}^n \to \mathbb{R}$ defined by $\overline{L}(x) = \frac{1}{2} x^T A x + b^T x + c$ is given by $\phi^\star(y) = \frac{1}{2}(y - b)^T A^{-1}(y - b) - c$. For example, see [4].

Going back to (B.1), we now deduce that this problem is equivalent to

$$\begin{aligned}\sup_{(\omega,y)\in\mathbb{R}^m_+\times\mathbb{R}^n} \quad & -\tfrac{1}{2}\left(y-\sum_{j=1}^m \omega_j g_{k,j}\right)^T W_k \left(y-\sum_{j=1}^m \omega_j g_{k,j}\right) \\ & +\sum_{j=1}^m \omega_j (f_{k,j}+g_{k,j}^T(x_k-x_{k,j})) - \delta_k\|y\|_\star \\ \text{s.t. } & \mathbb{1}^T\omega = 1.\end{aligned}$$

Letting $\gamma = -y$ and observing (2.18), this leads to (2.17), as desired.

Let us now show that the solution of problem (2.15) can be obtained from that of problem (2.17); in particular, we show that with (ω_k, γ_k) solving (2.17) and s_k defined in (2.5), the point x_{k+1} defined in (2.6) solves (2.15). First, optimality of the pair (ω_k, γ_k) with respect to (2.17) implies that, with the optimal $z_k \in \mathbb{R}$ for (2.16),

$$0 = -G_k^T W_k(G_k\omega_k+\gamma_k) + b_k - z_k\omega_k \tag{B.3a}$$

$$\text{and} \quad 0 \in -W_k(G_k\omega_k+\gamma_k) - \delta_k\partial\|\gamma_k\|_*. \tag{B.3b}$$

Using the fact that for any $\gamma \in \mathbb{R}^n$ one has

$$\partial\|\gamma\|_* = \{p\in\mathbb{R}^n : \|p\|\le 1 \text{ and } p^T\gamma = \|\gamma\|_*\},$$

it follows that there exists a vector $p_k \in \mathbb{R}^n$ such that

$$0 = W_k(G_k w_k+\gamma_k) + \delta_k p_k, \quad \|p_k\|\le 1, \quad \text{and} \quad p_k^T\gamma_k = \|\gamma_k\|_*. \tag{B.4}$$

Hence, evaluating the dual objective function at (ω_k, γ_k), one obtains

$$\begin{aligned} & -\tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k) + b_k^T\omega_k - \delta_k\|\gamma_k\|_* \\ = & -\tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k) + b_k^T\omega_k - \delta_k p_k^T\gamma_k \\ = & -\tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k) + b_k^T\omega_k + \gamma_k^T W_k(G_k\omega_k+\gamma_k) \\ = & \tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k) + b_k^T\omega_k - \omega_k^T G_k^T W_k(G_k\omega_k+\gamma_k). \end{aligned}$$

By duality theory, our desired conclusion follows as long as x_{k+1} is feasible for problem (2.15) and yields an objective function value equal to this dual objective function value. To see that this is the case, first notice that, by (B.4),

$$\|x_{k+1}-x_k\| = \|s_k\| = \|W_k(G_k\omega_k+\gamma_k)\| = \delta_k\|p_k\| \le \delta_k.$$

Second, observe that the primal objective value at x_{k+1} is

$$\begin{aligned} q_{k,m}(x_{k+1}) &= \tfrac{1}{2}(x_{k+1}-x_k)^T H_k(x_{k+1}-x_k) + z_k \\ &= \tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k) + b_k^T\omega_k - \omega_k^T G_k^T W_k(G_k\omega_k+\gamma_k), \end{aligned}$$

where the second equation follows from the definition of s_k in (2.5), the result of multiplying (B.3a) on the left by ω_k^T, and the fact that $\omega_k^T\omega_k = 1$.

All that remains is to prove the inequalities (2.19) and (2.21). Since x_k is feasible for (2.15) yielding an objective value of $l_{k,m}(x_k) = f(x_k)$, it follows that

$$0 \le f(x_k) - q_{k,m}(x_{k+1}) = f(x_k) - l_{k,m}(x_{k+1}) - \tfrac{1}{2}(x_{k+1}-x_k)^T H_k(x_{k+1}-x_k),$$

from which it follows that

$$\begin{aligned} f(x_k) - l_{k,m}(x_{k+1}) &\ge \tfrac{1}{2}(x_{k+1}-x_k)^T H_k(x_{k+1}-x_k) \\ &= \tfrac{1}{2}(G_k\omega_k+\gamma_k)^T W_k(G_k\omega_k+\gamma_k), \end{aligned}$$

yielding (2.19), as desired. Similarly, by weak duality it follows that

$$f(x_k) \geq -\tfrac{1}{2}(G_k\omega_k + \gamma_k)^T W_k(G_k\omega_k + \gamma_k) + b_k^T\omega_k - \delta_k\|\gamma_k\|_*,$$

which along with the fact that $l_{k,m}(x_{k+1}) = z_k$ and (B.3a) yields

$$\begin{aligned}
&f(x_k) - l_{k,m}(x_{k+1}) \\
&= f(x_k) - b_k^T\omega_k + \omega_k^T G_k W_k(G_k\omega_k + \gamma_k) \\
&\geq -\tfrac{1}{2}(G_k\omega_k + \gamma_k)^T W_k(G_k\omega_k + \gamma_k) - \delta_k\|\gamma_k\|_* + \omega_k^T G_k W_k(G_k\omega_k + \gamma_k) \\
&= \tfrac{1}{2}\omega_k^T G_k^T W_k G_k\omega_k - \tfrac{1}{2}\gamma_k^T W_k\gamma_k - \delta_k\|\gamma_k\|_*,
\end{aligned}$$

yielding (2.21), as desired.

C Self-Correcting Properties of BFGS Updating

The purpose of this appendix is to provide a proof of Theorem 3.1. First, observe that nonpositivity of the latter terms in (3.4) follows since $s_k \neq 0$ and $H_k \succ 0$ imply that $\cos^2\phi_k \in (0,1]$ and $\iota_k \in \mathbb{R}_{++}$ for all $k \in \mathbb{N}$, since $\ln(r) \leq 0$ for all $r \in (0,1]$, and since $1 - r + \ln(r) \leq 0$ for all $r \in \mathbb{R}_{++}$. Note also that

$$1 - r + \ln(r) \searrow -\infty \quad \text{as} \quad r \searrow 0 \quad \text{or} \quad r \nearrow \infty. \tag{C.1}$$

These facts are used explicitly in the following proof.

Theorem 3.1. For all $k \in \mathbb{N}$, define

$$\zeta_k := -\ln(\cos^2\phi_k) - \left(1 - \frac{\iota_k}{\cos^2\phi_k} + \ln\left(\frac{\iota_k}{\cos^2\phi_k}\right)\right) \geq 0. \tag{C.2}$$

Hence, from (2.9) and (3.4), it follows for all $k \in \mathbb{N}$ that

$$\psi(H_{k+1}) \leq \psi(H_k) + \theta - 1 - \ln\eta - \zeta_k,$$

from which it follows for all $K \in \mathbb{N}$ that

$$\psi(H_{K+1}) \leq \psi(H_1) + (\theta - 1 - \ln\eta)K - \sum_{k=1}^{K}\zeta_k.$$

Since, for all $K \in \mathbb{N}$, one has $\psi(H_{K+1}) \in \mathbb{R}_{++}$, this implies that

$$\frac{1}{K}\sum_{k=1}^{K}\zeta_k < \frac{1}{K}\psi(H_1) + (\theta - 1 - \ln\eta). \tag{C.3}$$

Now, considering fixed $p \in (0,1)$ and $K \in \mathbb{N}$, let $J_{p,K}$ be the set of indices corresponding to the $\lceil pK \rceil$ smallest elements of ζ_k for $k \in \{1,\ldots,K\}$, and let $\zeta_{p,K}$ denote the largest element of $\{\zeta_k\}_{k\in J_{p,K}}$. Then, it follows that

$$\frac{1}{K}\sum_{k=1}^{K}\zeta_k \geq \frac{1}{K}\left(\zeta_{p,K} + \sum_{k=1,k\notin J_{p,K}}^{K}\zeta_k\right) \geq \left(\frac{1}{K} + \frac{K - \lceil pK\rceil}{K}\right)\zeta_{p,K} \geq (1-p)\zeta_{p,K},$$

which along with (C.3) and the fact that $K \geq 1$ implies that, for all $k \in J_{p,K}$,

$$\zeta_k \leq \zeta_{p,K} < \frac{1}{1-p}(\psi(H_1) + \theta - 1 - \ln\eta) =: c_0 \in \mathbb{R}_{++}. \tag{C.4}$$

Since the facts that $\cos^2 \phi_k \in (0, 1]$ and $\iota_k \in \mathbb{R}_+$ for all $k \in \mathbb{N}$, (C.1), and (C.2) together imply that $\zeta_k \geq -\ln(\cos^2 \phi_k)$ for all $k \in J_{p,K}$, it follows from (C.4) that $-\ln(\cos^2 \phi_k) < c_0$ for all $k \in J_{p,K}$, which means that $\cos \phi_k > e^{-c_0/2} =: c_1 \in \mathbb{R}_{++}$ for all $k \in J_{p,K}$. That is, observing (3.3), the first inequality in (3.5) holds for any constant $\kappa \in (0, c_1]$. Now observe that (C.2), the fact that $-\ln(\cos^2 \phi_k) \geq 0$ for all $k \in \mathbb{N}$, and (C.4) imply for all $k \in J_{p,K}$ that

$$1 - \frac{\iota_k}{\cos^2 \phi_k} + \ln\left(\frac{\iota_k}{\cos^2 \phi_k}\right) > -c_0.$$

Hence, by (C.1), there exist $c_2 \in \mathbb{R}_{++}$ and $c_3 \in \mathbb{R}_{++}$ such that, for all $k \in J_{p,K}$,

$$c_2 \leq \frac{\iota_k}{\cos^2 \phi_k} \leq c_3.$$

Combining this with the fact (already proved) that $\cos \phi_k > c_1$ for all $k \in J_{p,K}$ and the fact that $\cos \phi_k \leq 1$ for all $k \in \mathbb{N}$, it follows, for all $k \in J_{p,K}$, that $c_1^2 c_2 \leq \iota_k \leq c_3$. Therefore, since $\|H_k s_k\|_2 / \|s_k\|_2 = \iota_k / \cos \phi_k$, it follows that

$$c_1^2 c_2 \leq \frac{\|H_k s_k\|_2}{\|s_k\|_2} \leq \frac{c_3}{c_1};$$

i.e., the latter inequalities in (3.5) hold for any $\sigma \in (0, c_1^2 c_2]$ and $\mu \in [c^{-1} c_3, \infty)$. □

D Proofs for SVANO-Bundle-Step

This appendix provides proofs for the results stated in §4.2 related to the SVANO-Bundle-Step algorithm. The results and proofs are based on those found in [41, §7.4], but modified to account for a variable-metric quadratic term in the subproblem objective rather than a proximal term with a fixed penalty parameter.

Lemma 4.2. Let $\tilde{x} \in \mathcal{X}_k$ be any point with $f(\tilde{x}) < f(x_k)$, the existence of which follows under the conditions of the lemma. Restricting the minimization on the right-hand side of (4.2) to the line segment $[x_k, \tilde{x}]$ and using convexity of f gives

$$\begin{aligned}
f_{H_k, \mathcal{X}_k}(x_k) &\leq \min_{x \in [x_k, \tilde{x}]} \; f(x) + \tfrac{1}{2}(x - x_k)^T H_k (x - x_k) \\
&= \min_{\Delta \in [0,1]} \; f((1 - \Delta)x_k + \Delta \tilde{x}) + \Delta^2 \tfrac{1}{2}(\tilde{x} - x_k)^T H_k (\tilde{x} - x_k) \\
&\leq \min_{\Delta \in [0,1]} \; (1 - \Delta) f(x_k) + \Delta f(\tilde{x}) + \Delta^2 \tfrac{1}{2}(\tilde{x} - x_k)^T H_k (\tilde{x} - x_k) \\
&= f(x_k) + \min_{\Delta \in [0,1]} \; \Delta(f(\tilde{x}) - f(x_k)) + \Delta^2 \tfrac{1}{2}(\tilde{x} - x_k)^T H_k (\tilde{x} - x_k).
\end{aligned}$$

Since $H_k \succ 0$, this last minimization problem over $\Delta \in [0, 1]$ involves a strongly convex quadratic function of Δ. Moreover, since $f(\tilde{x}) < f(x_k)$, the value of $\Delta \in [0, 1]$ that minimizes the function is strictly positive, meaning that the optimal value of the problem is strictly negative. The assertion of the lemma thus follows. □

Lemma 4.3. Let $(k, m) \in \mathbb{N} \times \mathbb{N}$ be given. Since $H_k \succ 0$, it follows by (2.14) that $l_{k,m}(x) \leq q_{k,m}(x)$ for all $x \in \mathbb{R}^n$, which implies the first inequality in (4.3). Moreover, $l_{k,m}$ being a pointwise underestimator of f throughout $\mathbb{R}^n$ means that $l_{k,m}(x) \leq f(x)$ for all $x \in \mathbb{R}^n$, which implies by (2.14) that $q_{k,m}(x) \leq f(x) + \frac{1}{2}(x - x_k)^T H_k (x - x_k)$ for all $x \in \mathbb{R}^n$. Letting $\hat{x}$ be the argument that solves the minimization problem in the right-hand-side of (4.2) for $\bar{x} = x_k$, it follows along with the arguments above that

$$q_{k,m}(x_{k,m+1}) \leq q_{k,m}(\hat{x}) \leq f(\hat{x}) + \tfrac{1}{2}(\hat{x} - x_k)^T H_k (\hat{x} - x_k) = f_{H_k, \mathcal{X}_k}(x_k),$$

which establishes the second inequality in (4.3). □

References

[1] J. Barzilai and J. M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., Third edition, 2006.

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Second edition, 1999.

[4] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, Nashua, NH, USA, 2009.

[5] C. G. Broyden. The Convergence of a Class of Double-Rank Minimization Algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6(1):76–90, 1970.

[6] J. V. Burke, A. S. Lewis, and M. L. Overton. A Robust Gradient Sampling Algorithm for Nonsmooth, Nonconvex Optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

[7] R. H. Byrd and J. Nocedal. A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization. *SIAM Journal on Numerical Analysis*, 26(3):727–739, 1989.

[8] R. H. Byrd, J. Nocedal, and Y. Yuan. Global Convergence of a Class of Quasi-Newton Methods on Convex Problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1189, 1987.

[9] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley & Sons, New York, NY, USA, 1983.

[10] F. E. Curtis. A Self-Correcting Variable-Metric Algorithm for Stochastic Optimization. In *International Conference on Machine Learning (ICML)*, 2016.

[11] F. E. Curtis and X. Que. An Adaptive Gradient Sampling Algorithm for Nonsmooth Optimization. *Optimization Methods and Software*, 28(6):1302–1324, 2013.

[12] F. E. Curtis and X. Que. A Quasi-Newton Algorithm for Nonconvex, Nonsmooth Optimization with Global Convergence Guarantees. *Mathematical Programming Computation*, DOI: 10.1007/s12532-015-0086-2, 2015.

[13] W. C. Davidon. Variable Metric Method for Minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[14] J. E. Dennis and J. J. Moré. A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods. *Mathematics of Computation*, 28(126):549–560, 1974.

[15] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics (SIAM), 1996.

[16] Y. Du and A. Ruszczynski. Rate of Convergence of the Bundle Method. arXiv:1609.00842, 2016.

[17] R. Fletcher. A New Approach to Variable Metric Algorithms. *Computer Journal*, 13(3):317–322, 1970.

[18] D. Goldfarb. A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computation*, 24(109):23–26, 1970.

[19] A. A. Goldstein. Optimization of lipschitz continuous functions. *Mathematical Programming*, 13(1):14–22, 1977.

[20] N. Haarala, K. Miettinen, and M. M. Mäkelä. New Limited Memory Bundle Method for Large-Scale Nonsmooth Optimization. *Optimization Methods and Software*, 19(6):673–692, 2004.

[21] N. Haarala, K. Miettinen, and M. M. Mäkelä. Globally Convergent Limited Memory Bundle Method for Large-Scale Nonsmooth Optimization. *Mathematical Programming*, 109(1):181–205, 2007.

[22] W. Hare and C. Sagastizábal. A Redistributed Proximal Bundle Method for Nonconvex Optimization. *SIAM Journal on Optimization*, 20(5):2442–2473, 2010.

[23] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. A Series of Comprehensive Studies in Mathematics. Springer-Verlag, New York, NY, USA, 1993.

[24] F. Kappel and A. V. Kuntsevich. An implementation of Shor's R-algorithm. *Computational Optimization and Applications*, 15(2):193–205, 2000.

[25] K. C. Kiwiel. A Linearization Algorithm for Nonsmooth Minimization. *Mathematics of Operations Research*, 10(2):185–194, 1985.

[26] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics. Springer-Verlag, New York, NY, USA, 1985.

[27] K. C. Kiwiel. A Method for Solving Certain Quadratic Programming Problems Arising in Nonsmooth Optimization. *IMA Journal of Numerical Analysis*, 6(2):137–152, 1986.

[28] K. C. Kiwiel. Restricted Step and Levenberg-Marquardt Techniques in Proximal Bundle Methods for Nonconvex Nondifferentiable Optimization. *SIAM Journal on Optimization*, 6(1):227–249, 1996.

[29] K. C. Kiwiel. Convergence of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization. *SIAM Journal on Optimization*, 18(2):379–388, 2007.

[30] C. Lemaráchal. Numerical Experiments in Nonsmooth Optimization. In E. A. Nurminski, editor, *Progress in Nondifferentiable Optimization*, pages 61–84. International Institute for Applied Systems Analysis (IIASA), 1982.

[31] A. S. Lewis and M. L. Overton. Nonsmooth Optimization via Quasi-Newton Methods. *Mathematical Programming*, 141(1):135–163, 2012.

[32] L. Lukšan and J. Vlček. A Bundle-Newton Method for Nonsmooth Unconstrained Minimization. *Mathematical Programming*, 83(1):373–391, 1998.

[33] R. Mifflin. *A modification and an extension of Lemarechal's algorithm for nonsmooth minimization*, pages 77–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

[34] R. Mifflin and C. Sagastizábal. A $\mathcal{VU}$-Algorithm for Convex Minimization. *Mathematical Programming*, 104(2):583–608, 2005.

[35] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Second edition, 2006.

[36] J. D. Pearson. Variable Metric Methods of Minimisation. *The Computer Journal*, 12(2):171–178, 1969.

[37] M. J. D. Powell. Some Global Convergence Properties of a Variable Metric Algorithm for Minimization with Exact Line Searches. In R. W. Cottle and C. E. Lemke, editors, *Nonlinear Programming, SIAM-AMS Proceedings, Vol. IX*. American Mathematical Society, 1976.

[38] M. J. D. Powell. Algorithms for Nonlinear Constraints that use Lagrangian Functions. *Mathematical Programming*, 14(1):224–248, 1978.

[39] K. Ritter. Local and Superlinear Convergence of a Class of Variable Metric Methods. *Computing*, 23(3):287–297, 1979.

[40] K. Ritter. Global and Superlinear Convergence of a Class of Variable Metric Methods. In H. König, B. Korte, and K. Ritter, editors, *Mathematical Programming at Oberwolfach*, pages 178–205. Springer Berlin Heidelberg, 1981.

[41] A. Ruszczynski. *Nonlinear Optimization*. Princeton University Press, 2006.

[42] D. F. Shanno. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

[43] N. Z. Shor. Minimization methods for non-differentiable functions. naukova dumka, kiev. *English translation published by Springer-Verlag, Berlin*, 1985.

[44] J. Werner. Über die globale Konvergenz von Variable-Metrik-Verfahren mit nicht-exakter Schrittweitenbestimmung. *Numerische Mathematik*, 31(3):321–334, 1978.