Secant penalized BFGS: a noise robust quasi-Newton method via penalizing the secant condition

Brian Irwin1 · Eldad Haber1

Received: 10 July 2021 / Accepted: 23 December 2022 / Published online: 9 January 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
In this paper, we introduce a new variant of the BFGS method designed to perform well when gradient measurements are corrupted by noise. We show that treating the secant condition with a penalty method approach motivated by regularized least squares estimation generates a parametric family with the original BFGS update at one extreme and not updating the inverse Hessian approximation at the other extreme. Furthermore, we find the curvature condition is relaxed as the family moves towards not updating the inverse Hessian approximation, and disappears entirely at the extreme where the inverse Hessian approximation is not updated. These developments allow us to develop a method we refer to as Secant Penalized BFGS (SP-BFGS) that allows one to relax the secant condition based on the amount of noise in the gradient measurements. SP-BFGS provides a means of incrementally updating the new inverse Hessian approximation with a controlled amount of bias towards the previous inverse Hessian approximation, which allows one to replace the overwriting nature of the original BFGS update with an averaging nature that resists the destructive effects of noise and can cope with negative curvature measurements. We discuss the theoretical properties of SP-BFGS, including convergence when minimizing strongly convex functions in the presence of uniformly bounded noise. Finally, we present extensive numerical experiments using over 30 problems from the CUTEst test problem set that demonstrate the superior performance of SP-BFGS compared to BFGS in the presence of both noisy function and gradient evaluations.

Keywords Quasi-Newton methods · Secant condition · Penalty methods · Least squares estimation · Measurement error · Noise robust optimization

1 Department of Earth, Ocean and Atmospheric Sciences, The University of British Columbia, Vancouver, BC, Canada
1 Introduction

Over the past 50 years, quasi-Newton methods have proved to be some of the most economical and effective methods for a variety of optimization problems. Originally conceived to provide some of the advantages of second order methods without the full cost of Newton’s method, quasi-Newton methods, which are also referred to as variable metric methods [26], are based on the observation that by differencing observed gradients, one can calculate approximate curvature information. This approximate curvature information can then be used to improve the speed of convergence, especially in comparison to first order methods, such as gradient descent. There are currently a variety of different quasi-Newton methods, with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [6, 13, 15, 42] almost certainly being the best known quasi-Newton method.

Modern quasi-Newton methods were developed for problems involving the optimization of smooth functions without constraints. The BFGS method is the best known quasi-Newton method because in practice it has demonstrated superior performance due to its very effective self-correcting properties [36]. Accordingly, BFGS has since been extended to handle box constraints [8], and shown to be effective even for some nonsmooth optimization problems [30]. Furthermore, a limited memory version of BFGS known as L-BFGS [32] has become a favourite algorithm for solving optimization problems with a very large number of variables, as it avoids directly storing approximate inverse Hessian matrices. However, BFGS and its relatives were not designed to explicitly handle noisy optimization problems, and noise can unacceptably degrade the performance of these methods.

The authors of [7] make the important observation that quasi-Newton updating is inherently an overwriting process rather than an averaging process. Fundamentally, differencing noisy gradients can produce harmful effects because the resulting approximate curvature information may be inaccurate, and this inaccurate curvature information may overwrite accurate curvature information. Newton’s method can naturally be viewed as a local rescaling of coordinates so that the rescaled problem is better conditioned than the original problem. Quasi-Newton methods attempt to perform a similar rescaling, but instead of using the (inverse) Hessian matrix to obtain curvature information for the rescaling, they use differences of gradients to obtain curvature information. Thus, it should be unsurprising that inaccurate curvature information obtained from differencing noisy gradients can be problematic because it means the resulting rescaling of the problem can be poor, and the conditioning of the rescaled problem could be even worse than the conditioning of the original problem.

With the above in mind, several works have dealt with how to improve the performance of quasi-Newton methods in the presence of noise. Many recent works focus on the empirical risk minimization (ERM) problem, which is ubiquitous in machine learning. For example, in [7] the authors propose a technique designed for the stochastic approximation (SA) regime that employs subsampled Hessian-vector products to collect curvature information pointwise and at spaced intervals, in contrast to the classical approach of computing the difference of gradients.
at each iteration. This work is built upon in [34], where the authors present a stochastic L-BFGS algorithm that draws upon the variance reduction approach of [25]. In [44], the authors outline a stochastic damped limited-memory BFGS (SdL-BFGS) method that employs damping techniques used in sequential quadratic programming (SQP). A stochastic block BFGS method that updates the approximate inverse Hessian matrix using a sketch of the Hessian matrix is proposed in [20]. Further work on stochastic L-BFGS algorithms, including convergence results, can be found in [11, 33, 41, 46].

Despite the importance of the ERM problem due to the current prevalence of machine learning, there are still a variety of important noisy optimization problems that arise in other contexts. In engineering design, numerical simulations are often employed in place of conducting costly, if even feasible, physical experiments. In this context, one tries to find optimal design parameters using the numerical simulation instead of physical experiments. Some examples from aerospace engineering, including interplanetary trajectory and wing design, can be found in [5, 12, 27]. Examples from materials engineering include stable composite design [1] and ternary alloy composition [21], amongst others [35], while examples from electrical engineering include power system operation [47], hardware verification [14], and antenna design [29]. Noise is often an unavoidable property of such numerical simulations, as the simulations can include stochastic internal components, and floating point arithmetic vulnerable to roundoff error. As optimizing noisy numerical simulations does not always fit the framework of the ERM problem, analyses of the behaviour of quasi-Newton methods in the presence of general bounded noise are of practical value when optimizing numerical simulations.

Apart from the analysis of the BFGS method with bounded errors in [45] and the noise tolerant versions of BFGS and L-BFGS developed in [43], there is relatively little work on the behaviour of quasi-Newton methods in the presence of general bounded noise. This work is most similar to [43], which builds upon the results of [45] to develop an extension of BFGS designed for the situation where unconstrained minimization must be performed using only function and gradient measurements corrupted by bounded noise. Although this paper considers the same situation as [43], the approach developed in this paper to address the corrupting effects of noise is distinct and potentially complementary to the approach used in [43]. While the approach of [43] employs a lengthening procedure that spaces out the points at which gradient differences are collected, in this paper we develop an approach based on relaxing the secant condition that does not require a lengthening procedure. Furthermore, [43] uses a modified version of a line search procedure based on the Armijo-Wolfe conditions. The approach developed in this paper uses a backtracking line search based on a relaxed Armijo condition.

1.1 Contributions

Noise is inevitably introduced into machine learning problems due to the approximations required to handle large datasets, and numerical simulations due to the effects of finite precision arithmetic, and parts of the simulator containing inherently
stochastic components. In this paper, we return to the fundamental theory underlying the design of quasi-Newton methods, which allows us to design a new variant of the BFGS method that explicitly handles the corrupting effects of noise. We do this as follows:

1. In Sect. 2, we review the setup and derivation of the original BFGS method.
2. In Sect. 3, motivated by regularized least squares estimation, we treat the secant condition of BFGS with a penalty method. This creates a new BFGS update formula that we refer to as Secant Penalized BFGS (SP-BFGS), which we show reduces to the original BFGS update formula in a limiting case, as expected.
3. In Sect. 4, we present an algorithmic framework for practically implementing SP-BFGS updating. We also discuss implementation details, including how to perform a line search and choose the penalty parameter in the presence of noise.
4. In Sect. 5, we discuss the theoretical properties of SP-BFGS, including how the penalty parameter influences the eigenvalues of the approximate inverse Hessian. This allows us to show that under appropriate conditions SP-BFGS iterations are guaranteed to converge linearly to a neighborhood of the global minimizer when minimizing strongly convex functions in the presence of uniformly bounded noise.
5. In Sect. 6, we study the empirical performance of SP-BFGS updating compared to BFGS updating and gradient descent by performing extensive numerical experiments with both convex and nonconvex objective functions corrupted by function and gradient noise. Results from a diverse set of over 30 problems from the CUTEst test problem set demonstrate that intelligently implemented SP-BFGS updating frequently outperforms BFGS updating and gradient descent in the presence of noise.
6. Finally, Sect. 7 concludes the paper and outlines directions for further work.

2 Mathematical background

In this section, as preliminaries to the main results of this paper, we review the setup and derivation of the original BFGS method.

2.1 BFGS setup

The BFGS method was originally designed to solve the following unconstrained optimization problem

$$\min_x \{ \phi(x) \}$$

with $x \in \mathbb{R}^n$, $\phi : \mathbb{R}^n \mapsto \mathbb{R}$, and $\phi$ being a smooth twice continuously differentiable and non-noisy function. Below, we use the notational conventions of [36], including $\phi_k = \phi(x_k)$. We begin by using the Taylor expansion of $\phi$ to build a local quadratic model $m_k$ of the objective function $\phi$ at the $k^{th}$ iterate $x_k$ of the optimization procedure.
\[
\phi(x_k + p) \approx \phi_k + \nabla \phi_k^T p + \frac{1}{2} p^T B_k p = m_k(p) \tag{2}
\]

where \( B_k \) is an \( n \times n \) symmetric positive definite matrix that approximates the Hessian matrix (i.e. \( B_k \approx \nabla^2 \phi_k \)). By setting the gradient of \( m_k \) to zero, we see that the unique minimizer \( p_k \) of this local quadratic model is

\[
p_k = -B_k^{-1} \nabla \phi_k \tag{3}
\]

and thus it is natural to update the next iterate \( x_{k+1} \) as

\[
x_{k+1} = x_k + \alpha_k p_k \tag{4}
\]

where \( \alpha_k \) is the step size along the direction \( p_k \), which is often chosen using a line search.

To avoid computing \( B_k \) from scratch at each iteration \( k \), we use the curvature information from recent gradient evaluations to update \( B_k \), and thus relatively economically form \( B_{k+1} \). A Taylor expansion of \( \nabla \phi \) reveals

\[
\nabla \phi(x_k + p) \approx \nabla \phi_k + \nabla^2 \phi_k p \tag{5}
\]

and so it is reasonable to require that the new approximate Hessian \( B_{k+1} \) satisfies

\[
\nabla \phi_{k+1} = \nabla \phi_k + \alpha_k B_{k+1} p_k \tag{6}
\]

which rearranges to

\[
B_{k+1} \alpha_k p_k = \nabla \phi_{k+1} - \nabla \phi_k . \tag{7}
\]

Now, define the two new quantities \( s_k \) and \( y_k \) as

\[
s_k := x_{k+1} - x_k = \alpha_k p_k , \tag{8a}
\]

\[
y_k := \nabla \phi_{k+1} - \nabla \phi_k . \tag{8b}
\]

Thus, we arrive at (9), which is known as the secant condition

\[
B_{k+1} s_k = y_k . \tag{9}
\]

In words, the secant condition dictates that the new approximate Hessian \( B_{k+1} \) must map the measured displacement \( s_k \) into the measured difference of gradients \( y_k \). If we denote the approximate inverse Hessian \( H_k = B_k^{-1} \approx \nabla^2 \phi_k^{-1} \), then the secant condition can be equivalently expressed as (10)

\[
H_{k+1} y_k = s_k . \tag{10}
\]

As \( H_{k+1} \) is not yet uniquely determined, to obtain the BFGS update formula, we impose a minimum norm restriction. Specifically, we choose \( H_{k+1} \) to be the solution of the following quadratic program over matrices
\[
\min_H \left\{ \frac{1}{2} \left\| W^{1/2}(H - H_k)W^{1/2} \right\|_F^2 \right\} \quad \text{s.t.} \quad H = H^T, \quad H y_k = s_k \tag{11}
\]

where \( \| \cdot \|_F \) denotes the Frobenius norm, and \( W^{1/2} \) the principal square root (see [24] or a similar reference) of a symmetric positive definite weight matrix \( W \) satisfying

\[
W s_k = y_k . \tag{12}
\]

As we will see, choosing the weight matrix \( W \) to satisfy (12) ensures that the resulting optimization method is scale invariant. The weight matrix \( W \) can be chosen to be any symmetric positive definite matrix satisfying (12), and the specific choice of \( W \) is not of great importance, as \( W \) will not appear directly in the main results of this paper. However, as a concrete example from [36], one could assume \( W = \bar{G}_k \), where \( \bar{G}_k \) is the average Hessian defined by

\[
\bar{G}_k = \int_0^1 \nabla^2 \phi(x_k + t\alpha p_k) dt . \tag{13}
\]

### 2.2 Solving for the BFGS update

To solve the quadratic program given by (11), we setup a Lagrangian \( \mathcal{L}(H, q, \Gamma) \) involving the constraints. Recalling that

\[
\left\| W^{1/2}(H - H_k)W^{1/2} \right\|_F^2 = \operatorname{Tr} \left( W(H - H_k)W(H - H_k)^T \right) , \tag{14}
\]

this gives the Lagrangian defined by (15) below

\[
\mathcal{L}(H, q, \Gamma) = \frac{1}{2} \operatorname{Tr} (W(H - H_k)W(H - H_k)^T) + \operatorname{Tr} (H y_k - s_k q^T) + \operatorname{Tr} (\Gamma (H - H^T)) \tag{15}
\]

where \( q \) is a vector of Lagrange multipliers associated with the secant condition, and \( \Gamma \) is a matrix of Lagrange multipliers associated with the symmetry condition. Taking the derivative of the Lagrangian \( \mathcal{L}(H, q, \Gamma) \) with respect to the matrix \( H \) yields

\[
\frac{\partial \mathcal{L}(H, q, \Gamma)}{\partial H} = W(H - H_k)W + q y_k^T + \Gamma^T - \Gamma \tag{16}
\]

and so we have the Karush-Kuhn-Tucker (KKT) system defined by the three equations (17a), (17b), and (17c) below

\[
W(H - H_k)W + q y_k^T + \Gamma^T - \Gamma = 0 , \tag{17a}
\]

\[
H y_k - s_k = 0 . \tag{17b}
\]

\[
H - H^T = 0 . \tag{17c}
\]
For brevity, we omit the details of the solution of the KKT system defined above because it is a limiting case of the system solved in Theorem 1. For an alternative geometric solution technique, we refer the interested reader to Section 2 of [22]. The minimizer $H^* = H_{k+1}$ is given by the well known BFGS update formula

$$
H_{k+1} = \left(I - \frac{s_k^T y_k}{s_k^T s_k}\right) H_k \left(I - \frac{y_k s_k^T}{y_k^T y_k}\right) + \frac{s_k s_k^T}{y_k^T y_k}
$$

which, if we define the curvature parameter $\rho_k = \frac{1}{s_k^T y_k}$, can be equivalently written as

$$
H_{k+1} = \left(I - \rho_k s_k s_k^T\right) H_k \left(I - \rho_k y_k y_k^T\right) + \rho_k s_k s_k^T.
$$

Applying the Sherman-Morrison-Woodbury formula (see [23]) to the BFGS update formula immediately above, one can also write the BFGS update in terms of the approximate Hessian $B_k = H_k^{-1}$ instead of the approximate inverse Hessian. Again, for brevity, the details are omitted because they are a special case of Theorem 2 shown later. The result is

$$
B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T B_k s_k} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \rho_k y_k y_k^T.
$$

To ensure the updated approximate Hessian $B_{k+1}$ is positive definite, we must enforce that

$$
s_k^T B_{k+1} s_k > 0.
$$

Substituting $B_{k+1} s_k = y_k$ from the secant condition, the condition (21) becomes

$$
s_k^T y_k > 0
$$

which is known as the curvature condition, as it is equivalent to

$$
\frac{1}{\rho_k} > 0.
$$

### 3 Derivation of secant penalized BFGS

In this section, having reviewed the construction of the original BFGS method, we now show how treating the secant condition with a penalty method approach motivated by regularized least squares estimation allows one to generalize the original BFGS update.
3.1 Penalizing the secant condition

By applying a penalty method (see Chapter 17 of [36]) to the secant condition instead of directly enforcing the secant condition as a constraint, we obtain the problem

\[
\min_H \left\{ \frac{1}{2} \left\| W^{1/2} (H - H_k) W^{1/2} \right\|^2_F + \frac{\beta_k}{2} \left\| W^{1/2} (H y_k - s_k) \right\|^2_2 \right\} \quad \text{s.t.} \quad H = H^T
\]

(24)

where \(\beta_k \in [0, +\infty] \) is a penalty parameter that determines how strongly to penalize violations of the secant condition. As we will see, one recovers the solution to the constrained problem (11) in the limit \(\beta_k = +\infty\), so \(\beta_k\) can be intuitively thought of as the cost of violating the secant condition. By treating the symmetry constraint with a matrix \(\Gamma\) of Lagrange multipliers again, we obtain the following Lagrangian

\[
\mathcal{L}(H, \Gamma) = \frac{1}{2} \text{Tr} \left( W (H - H_k) W (H - H_k)^T \right) + \frac{\beta_k}{2} \left\| W^{1/2} (H y_k - s_k) \right\|^2 + \text{Tr} \left( \Gamma (H - H^T) \right).
\]

(25)

Defining the residual associated with the secant condition as \(r_k(H) := H y_k - s_k\) and \(u := \beta_k W r_k\), the first order optimality conditions of (25) can be written as the system

\[
W (H - H_k) W + u y_k^T + \Gamma^T - \Gamma = 0,
\]

(26a)

\[
H y_k - s_k - \frac{W^{-1} u}{\beta_k} = 0,
\]

(26b)

\[
H - H^T = 0.
\]

(26c)

Note that, as expected, in the limit \(\beta_k = +\infty\), the system given by (26a), (26b), and (26c) reduces to the KKT system given by (17a), (17b), and (17c).

We now find an explicit closed form solution to the problem given by (24), which is given in Theorem 1.

**Theorem 1** (SP-BFGS Update) The update formula given by the minimizer \(H^*\) of the problem defined by (24), which can be obtained by solving the system given by (26a), (26b), and (26c), is the SP-BFGS update

\[
H_{k+1} = \left( I - \omega_k s_k y_k^T \right) H_k \left( I - \omega_k y_k s_k^T \right) + \omega_k \left[ \frac{y_k}{\omega_k} + (\gamma_k - \omega_k) y_k^T H_k y_k \right] s_k s_k^T
\]

(27)

where

\[
\gamma_k = \frac{1}{(s_k^T y_k + \frac{1}{\beta_k})}, \quad \omega_k = \frac{1}{(s_k^T y_k + \frac{2}{\beta_k})}.
\]

(28)

**Proof** See Appendix 1.
At this point, a few comments are in order regarding the SP-BFGS update given by (27). First, observe that as $\beta_k \to +\infty$, we have that $\omega_k \to \rho_k$ and $\gamma_k \to \rho_k$. As a result, when $\beta_k = +\infty$, one recovers the original BFGS update, as expected. Second, also observe that as $\beta_k \to 0$, we have that $\omega_k \to 0$ and $\gamma_k \to 0$. As a result, we see that in the case $\beta_k = 0$ the SP-BFGS update reduces to $H_{k+1} = H_k$. This is again expected because as $\beta_k \to 0$, the cost of violating the secant condition goes to zero, and the minimum norm symmetric update is simply $H_{k+1} = H_k$.

We now examine what the analog of the curvature condition (22) is for SP-BFGS. Lemma 1 demonstrates that (29) is the SP-BFGS analog of the BFGS curvature condition (22).

**Lemma 1** (Positive Definiteness Of SP-BFGS Update) If $H_k$ is positive definite, then the $H_{k+1}$ given by the SP-BFGS update (27) is positive definite if and only if the SP-BFGS curvature condition

\[ s_T^T y_k > -\frac{1}{\beta_k} \]  

(29)

is satisfied.

**Proof** See Appendix 2. \(\square\)

The result in Lemma 1 warrants some discussion. First, the limiting behaviour with respect to $\beta_k$ is consistent with Theorem 1. As $\beta_k \to +\infty$, condition (29) reduces to the BFGS curvature condition (22). As $\beta_k \to 0$, condition (29) reduces to no condition at all, as $s_T^T y_k > -\infty$ is always true. This is consistent with the observation that when $\beta_k = 0$, the minimum norm symmetric update is $H_{k+1} = H_k$, and in this case $H_{k+1}$ is guaranteed to be positive definite if $H_k$ is positive definite, regardless of $s_T^T y_k$.

From the proof of Lemma 1 (see (102)), it is now clear that

\[ y^T_k H_{k+1} y_k = \left( \frac{\beta_k y^T_k s_k}{1 + \beta_k y^T_k s_k} \right) y^T_k s_k + \left( \frac{1}{1 + \beta_k y^T_k s_k} \right) y^T_k H_k y_k \]  

(30)

and so $y^T_k H_{k+1} y_k$ is a convex combination of $y^T_k s_k$ and $y^T_k H_k y_k$. Thus, as $\beta_k$ varies from 0 to $+\infty$, $H_{k+1}$ moves from the current inverse Hessian approximation $H_k$ at one extreme ($\beta_k = 0$) to the original BFGS update at the other extreme ($\beta_k = +\infty$). As $\beta_k$ decreases towards 0, $H_{k+1}$ is increasingly biased towards the current approximation $H_k$. From a regularized least squares estimation perspective, $\beta_k$ plays the role of a regularization parameter that controls the amount of bias in the estimate of $H_{k+1}$. Note that this behaviour is somewhat similar to the behaviour of Powell damping [39], although Powell damping was introduced to handle approximating a potentially indefinite Hessian of the Lagrangian in constrained optimization problems, and not noise.

We finish introducing the SP-BFGS update by applying the Sherman-Morrison-Woodbury formula to (27), which allows us to write the update in terms of the
approximate Hessian $B_k$ instead of the approximate inverse Hessian $H_k$. The result is given in Theorem 2.

**Theorem 2** (SP-BFGS Inverse Update) The SP-BFGS update formula given by (27) can be written in terms of $B_k = H_k^{-1}$ as

$$
\beta_{k+1} = \beta_k - \frac{\alpha_k \left[ (\alpha_k - \beta_k) B_k^{-1} y_k - \frac{\beta_k}{\alpha_k} B_k y_k + (1 - \alpha_k) \beta_k \gamma_k + \alpha_k \gamma_k \beta_k \gamma_k \right]}{\left( (\alpha_k - \beta_k) B_k^{-1} y_k - \frac{\beta_k}{\alpha_k} (\alpha_k \beta_k \gamma_k) - (1 - \alpha_k)^2 \right)}. 
$$

**Proof** See Appendix 3. \qed

Note that the limiting behaviour of Theorem 2 with respect to $\beta_k$ is again consistent. When $\beta_k = +\infty$, we obtain the original BFGS inverse update (20), and when $\beta_k = 0$, we obtain $B_{k+1} = B_k$. One complication with respect to the SP-BFGS inverse update (107) is that $B_{k+1}$ cannot in general be expressed solely in terms of $B_k$ due to the presence of $y_k^T B_k^{-1} y_k$ (i.e. $y_k^T H_k y_k$) in the denominator.

### 4 Algorithmic framework

We now outline how to practically implement SP-BFGS updating. We consider the situation where one has access to noise corrupted versions of a smooth function $\phi$ and its gradient $\nabla \phi$ that can be decomposed as

$$
f(x) = \phi(x) + e(x), \quad (31)$$

$$
g(x) = \nabla \phi(x) + e(x). \quad (32)
$$

In (31) and (32), $\phi$ is a smooth twice continuously differentiable function as in Sect. 2.1, and $e(x)$ is a scalar representing noise in the function evaluations. Similarly, $\nabla \phi$ is the gradient of the smooth function $\phi$, while $e(x)$ is a vector representing noise in the gradient evaluations. Similar decompositions are used in [2, 14, 45].

#### 4.1 Minimization routine

Algorithm 1 outlines a general procedure for minimizing a noisy function with noisy function and gradient values $f$ and $g$ that can be decomposed as shown in (31) and (32). The inputs to the procedure in Algorithm 1 are a means of evaluating the noisy objective function $f(x)$ and gradient $g(x)$, the starting point $x^0$, and an initial inverse Hessian approximation $H^0$. As the best convergence/stopping test is problem dependent, we note that standard gradient and function value based tests can be employed in conjunction with smoothing and noise estimation techniques (e.g. see Section 3.3.4 of [2]). In the next subsections, we discuss how to choose the step size $\alpha_k$ and the penalty parameter $\beta_k$. 
4.2 Choosing the step size $\alpha_k$

Classically, during BFGS updating $\alpha_k$ is chosen to satisfy the Armijo-Wolfe conditions. As function and gradient evaluations are not corrupted by noise in the classical BFGS setting, we can write the Armijo condition, also known as the sufficient decrease condition, as

$$\phi_{k+1} \leq \phi_k + c_1 \alpha_k \nabla \phi_k^T p_k$$  \hspace{1cm} (33)

and the Wolfe condition, also known as the curvature condition, as

$$\nabla \phi_{k+1}^T p_k \geq c_2 \nabla \phi_k^T p_k$$  \hspace{1cm} (34)

where $0 < c_1 < c_2 < 1$, with well known choices being $c_1 = 10^{-4}$ and $c_2 = 0.9$. Observe that by adding $\nabla \phi_k^T p_k$ to both sides of (34) and multiplying by $\alpha_k$, (34) becomes

$$\gamma_k^T s_k = [\nabla \phi_{k+1} - \nabla \phi_k]^{T} \alpha_k p_k \geq (c_2 - 1) \nabla \phi_k^T \alpha_k p_k .$$  \hspace{1cm} (35)

If $p_k$ is a descent direction then $\nabla \phi_k^T p_k < 0$, and combined with $(c_2 - 1) < 0$ and $\alpha_k > 0$, one sees that (35) implies
so (34) effectively enforces (22) when no gradient noise is present.

In the presence of noisy gradients, we argue that in general it no longer makes sense to enforce the Wolfe condition (34). In the presence of gradient noise, (34) becomes

\[ [\nabla \phi_{k+1} + e_{k+1}]^T p_k \geq c_2 [\nabla \phi_k + e_k]^T p_k \]

which can behave erratically once the noise vectors \( e_{k+1} \) and \( e_k \) start to dominate the gradient of \( \phi \). For example, the noise vectors \( e_{k+1} \) and \( e_k \) can cause both sides of (37) to erratically change sign, in which case whether or not the Wolfe condition is satisfied can be governed by randomness more than anything else.

We argue that because the SP-BFGS update allows one to relax the curvature condition based on the value of \( \beta_k \) as shown in the SP-BFGS curvature condition (29), it is appropriate to drop the Wolfe condition entirely in the presence of gradient noise and instead employ only a version of the sufficient decrease condition when choosing \( \alpha_k \). In the situation where gradient noise is present but function noise is not (i.e. \( f(x) = \phi(x) \) in (31)), one can use a backtracking line search based on the sufficient decrease condition, which can guarantee convergence to a neighborhood of a stationary point of \( \phi \). The situation where noise is present in both function and gradient evaluations is trickier. Similar to the approach presented in Section 4.2 of [2], one option is to use a backtracking line search with a relaxed sufficient decrease condition of the form

\[ f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k g_k^T p_k + 2 \epsilon_A \]

where \( \epsilon_A \geq 0 \) is a noise tolerance parameter and \( p_k = -H_k g_k \). In Theorem 4.2 of [2], the authors show that under Assumptions 1 and 2, using the iteration (4) and a backtracking line search governed by the relaxed Armijo condition (38) with \( p_k = -g_k \) guarantees linear convergence to a neighborhood of the global minimizer for strongly convex functions.

**Assumption 1 (Uniform Function Noise Bound)** There exists a nonnegative constant \( \tilde{\epsilon}_f \geq 0 \) such that

\[ |f(x) - \phi(x)| = |e(x)| \leq \tilde{\epsilon}_f , \quad \forall x \in \mathbb{R}^n . \]  

**Assumption 2 (Uniform Gradient Noise Bound)** There exists a nonnegative constant \( \tilde{\epsilon}_g \geq 0 \) such that

\[ \|g(x) - \nabla \phi(x)\|_2 = \|e(x)\|_2 \leq \tilde{\epsilon}_g , \quad \forall x \in \mathbb{R}^n . \]

In Sect. 5, Theorem 5 provides a type of extension of Theorem 4.2 of [2] to a quasi-Newton iteration with positive definite \( H_k \). A quasi-Newton extension of Theorem 4.2 of [2] is relevant to SP-BFGS updating because, as we will formally see in Sect. 5.1, control of \( \beta_k \) makes it possible to uniformly bound the minimum and maximum eigenvalues of \( H_{k+1} \).
4.3 Choosing the penalty parameter $\beta_k$

As the choice of $\beta_k$ determines how strongly to bias the estimate of $H_{k+1}$ towards $H_k$, the choice of $\beta_k$ is fundamentally connected to the amount of noise present in the measured gradients $g_{k+1}$ and $g_k$. In brief, if the amount of noise present in the measured gradients is large, $\beta_k$ should be small to avoid overfitting the noise, and if the amount of noise present in the measured gradients is small, $\beta_k$ should be large to avoid underfitting curvature information. We explore this point in more detail below.

To gain some intuition regarding how gradient differencing behaves in the presence of noise, note that as $\nabla \phi(x)$ is continuous, for each $k \geq 0$ one has

$$\lim_{\alpha_k \downarrow 0} \left\| \nabla \phi(x_k + \alpha_k p_k) - \nabla \phi(x_k) \right\|_2 = 0. \quad (41)$$

However, even if $\lim_{\alpha_k \downarrow 0} [g(x_k + \alpha_k p_k)]$ exists, due to noise one cannot in general guarantee

$$\lim_{\alpha_k \downarrow 0} \left[ g(x_k + \alpha_k p_k) \right] - g(x_k) = 0. \quad (42)$$

Assuming $\lim_{\alpha_k \downarrow 0} [g(x_k + \alpha_k p_k)]$ exists, using the continuity of $\nabla \phi(x)$, Assumption 2, and the triangle inequality, one can conclude that

$$0 \leq \left\| \lim_{\alpha_k \downarrow 0} [g(x_k + \alpha_k p_k)] - g(x_k) \right\|_2 \leq 2\bar{e}_g. \quad (43)$$

As a result, it is now clear that in the presence of uniformly bounded gradient noise, sending the step size $\alpha_k$ to zero, and thus $s_k$ to zero, only bounds the difference of measured gradients within a ball with radius dependent on the gradient noise bound $\bar{e}_g$. Clearly, the difference of gradient measurements taken at the same point will always lie within a ball with radius $2\bar{e}_g$ even if $\lim_{\alpha_k \downarrow 0} [g(x_k + \alpha_k p_k)]$ does not exist, as the bound in Assumption 2 holds everywhere.

As $g_{k+1}$ and $g_k$ can be decomposed into smooth and noise components, so can $s_k^T y_k$, giving

$$s_k^T y_k = s_k^T y_k^{\text{smooth}} + s_k^T y_k^{\text{noise}} = s_k^T [\nabla \phi_{k+1} - \nabla \phi_k] + s_k^T [e_{k+1} - e_k]. \quad (44)$$

In conjunction with the Cauchy-Schwarz inequality, Assumption 2 implies that

$$-2\bar{e}_g \|s_k\|_2 \leq s_k^T [e_{k+1} - e_k] \leq 2\bar{e}_g \|s_k\|_2 \quad (45)$$

and so we have the lower and upper bounds

$$-2\bar{e}_g \|s_k\|_2 + s_k^T y_k^{\text{smooth}} \leq s_k^T y_k \leq s_k^T y_k^{\text{smooth}} + 2\bar{e}_g \|s_k\|_2. \quad (46)$$

From (46), it is clear that the bound on the effect of the noise grows linearly with $\|s_k\|_2$. However, by using the average Hessian $\tilde{G}_k$ from (13) and applying Taylor’s theorem to $\nabla \phi$, it is also clear that
\[ s_k^T y^\text{smooth} = s_k^T \bar{G}_k s_k = \mathcal{O}(\|s_k\|^2) \]  

and so

\[ s_k^T y_k = \mathcal{O}(\|s_k\|^2) + \mathcal{O}(\|s_k\|) \]  

where the \( \mathcal{O}(\|s_k\|^2) \) term is due to the true curvature of the smooth function \( \phi \), and the \( \mathcal{O}(\|s_k\|) \) term is due to noise. Thus, we have now illustrated an important general behaviour given Assumption 2. As \( \|s_k\|^2 \) dominates \( \|s_k\| \) as \( \|s_k\| \to 0 \), the effects of noise can dominate the true curvature for small \( s_k \). Conversely, as \( \|s_k\|^2 \) dominates \( \|s_k\| \) as \( \|s_k\| \to +\infty \), the true curvature can dominate the effects of noise for large \( s_k \).

### 4.3.1 Models for initial choice of \( \beta_k \)

Given the above analysis, a simple model for producing an initial choice of \( \beta_k \) is to make \( \beta_k \) grow linearly with \( \|s_k\| \), such as

\[ \beta_k = N_s \|s_k\| \]  

where \( N_s > 0 \) is a slope parameter. As \( \|s_k\| \to 0 \), \( H_{k+1} \to H_k \), which is desirable because the effects of noise likely dominate as \( \|s_k\| \to 0 \). Increasingly biasing the estimate of \( H_{k+1} \) towards \( H_k \) reduces how much \( H_{k+1} \) can be corrupted by noise, and relaxes the SP-BFGS curvature condition (29). Also, as shown earlier, because \( \nabla \phi \) is continuous, the true difference of gradients is guaranteed to go to zero as \( s_k \) approaches zero. As a result, without noise present, it is natural that \( H_{k+1} \to H_k \) as \( s_k \to 0 \). In the presence of noise, we wish for this behaviour to be preserved. Informally, one can intuitively think of wanting \( H_k \) to behave as an approximate average inverse Hessian, and the averaging should remove the corrupting effects of noise, leaving \( H_k \) to behave as if no noise were present. Similarly, as \( \|s_k\| \to +\infty \), \( \beta_k \to +\infty \), and one recovers the BFGS update in the limit, which is desirable because the effects of noise are likely dominated by the true curvature as \( \|s_k\| \to +\infty \). The slope parameter \( N_s \) dictates how sensitive \( \beta_k \) is to \( \|s_k\| \), and should be set proportional to the gradient noise level (i.e. \( \bar{\epsilon}_g \)). Intuitively, if the gradient noise level is low, \( \beta_k \) should grow quickly with \( \|s_k\| \), as the effect of noise diminishes quickly, and vice versa.

It may also be desirable to modify (49) to

\[ \beta_k = \max \left\{ N_s \|s_k\| - N_o, 0 \right\} \]

where \( N_o > 0 \) is an intercept parameter. The inclusion of \( N_o \) allows one to stop updating \( H_k \) if \( \|s_k\| \) is sufficiently small. For example, it may be desirable to stop updating \( H_k \) when one is very close to a stationary point, as gradient measurements are likely heavily dominated by noise.
4.3.2 Ensuring the SP-BFGS curvature condition (29) Is Satisfied

The initial choice of $\beta_k$ produced by (49) or (50) may not satisfy the SP-BFGS curvature condition (29). If the initial choice of $\beta_k$ does not satisfy condition (29), then the user must adjust the choice of $\beta_k$ so that the new choice of $\beta_k$ does satisfy condition (29) before proceeding. As $\beta_k$ is chosen after $s_k$ and $y_k$ are calculated in Algorithm 1, one can solve for all $\beta_k$ values satisfying (29), yielding

$$\beta_k \in \begin{cases} [0, +\infty) & \text{if } s_k^Ty_k > 0 \\ [0, +\infty) & \text{if } s_k^Ty_k = 0 \\ \left[0, -1/s_k^Ty_k \right) & \text{if } s_k^Ty_k < 0 \end{cases}. \quad (51)$$

In the classical BFGS scenario where no gradient noise is present, the curvature condition (22) may fail if $\alpha_k$ is not chosen based on the Armijo-Wolfe conditions and $\phi$ is not strongly convex. In practice, one of the most common strategies to handle condition (22) not being satisfied is to skip the BFGS update (i.e. set $H_{k+1} = H_k$) when this occurs, which corresponds to an SP-BFGS update with the choice $\beta_k = 0$. Thus, in the SP-BFGS context, BFGS with update skipping corresponds to switching between setting $\beta_k = +\infty$ if $s_k^Ty_k > 0$ and $\beta_k = 0$ if $s_k^Ty_k \leq 0$. However, this simple strategy has the downside of potentially producing poor inverse Hessian approximations if updates are skipped too frequently.

Unlike in the classical BFGS scenario, with SP-BFGS updating, curvature information can be incorporated even if the measured curvature $s_k^Ty_k$ is negative. In addition to having the option of skipping updates when $s_k^Ty_k < 0$, one can also alternatively relax the SP-BFGS curvature condition (29) by picking an acceptable value of $\beta_k$ from the set shown in (51). For example, one could set $\beta_k = -c/s_k^Ty_k$ for some choice of $0 \leq c < 1$ if the initial choice of $\beta_k$ does not satisfy (29). Unless $c = 0$, such an approach never entirely skips incorporating measured curvature information, but instead weights how heavily the measured curvature information affects $H_{k+1}$.

5 Convergence of SP-BFGS

In this section, we discuss relevant theoretical and convergence properties of SP-BFGS. First, it is important to note that for specific choices of the sequence of penalty parameters $\beta_k$, known convergence results already exist. Specifically, if $\beta_k = +\infty$ for all $k$, then SP-BFGS updating is equivalent to BFGS updating. Although there are not many works on the convergence properties of BFGS updating in the presence of uniformly bounded noise, such as in Assumptions 1 and 2, in [45] and [43] the authors provide convergence results for BFGS variants that employ Armijo-Wolfe conditions based line searches and lengthen the gradient differencing interval in the presence of uniformly bounded function and gradient noise. At the other extreme, if $\beta_k = 0$ for all $k$, then one obtains a scaled gradient method for general $H^0 > 0$, and this becomes the gradient method when $H^0 = I$. Convergence analyses of the
gradient method in the presence of uniformly bounded function and gradient noise for both a fixed step size and backtracking line search are provided in Section 4 of [2].

Given that perhaps the defining feature of SP-BFGS updating is the ability to vary $\beta_k$ at each iteration, we focus our attention on how varying $\beta_k$ can influence convergence behaviour in this section. As a result, most of the ensuing analysis centers around situations where the condition number of $H_k$ can be bounded. We do not employ the approach of bounding the cosine of the angle between the update direction $p_k$ and the negative gradient above zero, and then showing that the condition number of $H_k$ is bounded, which is similar to the approaches taken when no noise is present in [9, 10], and when noise is present in [45]. Although it may be possible to apply the strategies employed in [9, 10, 45] to establish convergence results for SP-BFGS, such an analysis is complicated enough that it is not the approach taken in this initial paper.

5.1 The influence of $\beta_k$ on $H_{k+1}$

We first examine how $\beta_k$ determines how much the maximum and minimum eigenvalues $\lambda_{\text{max}}(H_{k+1})$ and $\lambda_{\text{min}}(H_{k+1})$ can change. In what follows, $\lambda(H)$ denotes the set of eigenvalues $\lambda_1, \ldots, \lambda_n$ of the matrix $H \in \mathbb{R}^{n \times n}$. We provide upper bounds on $\lambda_{\text{max}}(B_{k+1})$ and $\lambda_{\text{max}}(H_{k+1})$ via Theorem 3. As $H_k = B_k^{-1}$, $1/\lambda_{\text{min}}(H_{k+1}) = \lambda_{\text{max}}(B_{k+1})$, and putting an upper bound on $\lambda_{\text{max}}(B_{k+1})$ is equivalent to putting a lower bound on $\lambda_{\text{min}}(H_{k+1})$.

**Theorem 3** (Eigenvalue Upper Bounds) When $H_{k+1}$ is given by the SP-BFGS update (27), the following upper bounds (52) and (53) hold

$$\lambda_{\text{max}}(H_{k+1}) < \text{Tr}(H_{k+1}) \leq \left( 1 + \gamma_k \|y_k\|_2 \|s_k\|_2 \right)^2 \text{Tr}(H_k) + \gamma_k \|s_k\|_2^2, \quad (52)$$

$$\lambda_{\text{max}}(B_{k+1}) < \text{Tr}(B_{k+1}) \leq \left( 1 + \beta_k \|y_k\|_2 \|s_k\|_2 \right) \text{Tr}(B_k) + \gamma_k \|y_k\|_2^2. \quad (53)$$

**Proof** See Appendix 4. \qed

With Theorem 3 in hand, observe that because $s_k, y_k, H_k,$ and $B_k$ are known before the value of $\beta_k$ is chosen in Algorithm 1, and because $y_k$ depends only on $\beta_k$ if the values of $s_k$ and $y_k$ are already known (i.e. fixed), one can directly control the value of the right sides of (52) and (53) because the right sides of both expressions above only depend on $\beta_k$ now. As a result, at each iteration $k$ it is possible to directly calculate the values of $\beta_k$ that satisfy...
and

$$\left( 1 + \frac{\|y_k\|_2\|s_k\|_2}{(s_k^T y_k + \frac{1}{\beta_k})} \right)^2 \text{Tr}(H_k) + \frac{\|s_k\|_2^2}{(s_k^T y_k + \frac{1}{\beta_k})} \leq C_H^k$$

for known constants $0 < \text{Tr}(H_k) < C_H^k$ and $0 < \text{Tr}(B_k) < C_B^k$, which allows one to bound the growth of the maximum eigenvalues at each iteration $k$. Standard BFGS updating corresponds to setting $\beta_k = +\infty$ for all $k$, and as this is the largest possible value of $\beta_k$, one can no longer formally guarantee that $\lambda_{\text{max}}(B_{k+1})$ and $\lambda_{\text{max}}(H_{k+1})$ are bounded from above at each iteration because the measured curvature $s_k^T y_k$ may become arbitrarily close to zero due to the effects of noise. The key takeaway is that upper bounds on $\lambda_{\text{max}}(H_{k+1})$ and $\lambda_{\text{max}}(B_{k+1})$ can be tightened arbitrarily close to $\text{Tr}(H_k)$ and $\text{Tr}(B_k)$ by shrinking $\beta_k$ towards zero.

Thus, if one must enforce a bound of the form $\lambda_{\text{max}}(H_{k+1}) < C_H$ and a bound of the form $\lambda_{\text{max}}(B_{k+1}) < C_B$ for all $k \geq 0$, where $0 < \text{Tr}(H_0) < C_H$ and $0 < \text{Tr}(B_0) < C_B$ are positive constants, there exist nontrivial sequences of sufficiently small $\beta_k$ with $\lim_{k \to \infty} \beta_k = 0$ that ensure the bounds hold for all $k$. To see this, observe that if one sets all the per iteration growth bounds $C_H^k$ and $C_B^k$ so that $\lambda_{\text{max}}(H_0) < C_H^0 \leq C_H^1 \leq \cdots < C_H$ and $\lambda_{\text{max}}(B_0) < C_B^0 \leq C_B^1 \leq \cdots < C_B$, choices of $\beta_k$ from the resulting acceptable values of $\beta_k$ at each iteration $k$ can be combined to make a sequence of $\beta_k$ that satisfies the bounds $\lambda_{\text{max}}(H_{k+1}) < C_H$ and $\lambda_{\text{max}}(B_{k+1}) < C_B$ for all $k$. As there are many possible choices of per iteration growth bounds that satisfy $C_H^0 \leq C_H^1 \leq \cdots < C_H$ and $C_B^0 \leq C_B^1 \leq \cdots < C_B$, the sequence of $\beta_k$ that satisfies the bounds $\lambda_{\text{max}}(H_{k+1}) < C_H$ and $\lambda_{\text{max}}(B_{k+1}) < C_B$ is nonunique in general. Some choices of per iteration growth bounds have possible sequences of $\beta_k$ that satisfy $\beta_k = 0$ for all $k \geq K$, where $K$ is a positive integer, and some choices of per iteration growth bounds have possible sequences of $\beta_k$ where $\beta_k$ instead only approaches zero in the limit $k \to \infty$.

### 5.2 Minimization of strongly convex functions

Having established that SP-BFGS iterations can maintain bounds on the maximum and minimum eigenvalues of the approximate inverse Hessians via sufficiently small choices of $\beta_k$, we now consider minimizing strongly convex functions in the presence of bounded noise. We introduce Assumption 3, and the notation $x^*$ to denote the argument of the unique minimum of $\phi$, and $\phi^* = \phi(x^*)$ to denote the minimum.

**Assumption 3** *(Strong Convexity Of $\phi$)* The function $\phi \in C^2$ is twice continuously differentiable and there exist positive constants $0 < m \leq M$ such that

$$mI \preceq \nabla^2 \phi(x) \preceq MI, \quad \forall x \in \mathbb{R}^n.$$  

(66)
Lemma 2 establishes a family of regions where the inner product $\nabla \phi_k^T H_k g_k$ is bounded. Thus, Lemma 2 with the choice $\xi = 0$ can be used to establish a region where $H_k g_k$ may not provide a descent direction with respect to $\phi$ due to noise dominating gradient measurements. Outside of this region, $H_k g_k$ is guaranteed to provide a descent direction for $\phi$. Lemma 3 establishes conditions for the inequality $\nabla \phi_k^T H_k g_k \geq \varepsilon \nabla \phi_k^T g_k$ to be true for $\varepsilon > 0$.

**Lemma 2** (Regions Where Gradient Noise Can Dominate $\nabla \phi$) Suppose Assumptions 2 and 3, and the decomposition in (32) apply. Let $H$ be a symmetric positive definite matrix bounded by $\Psi I \leq H \leq \Psi I$, where $0 < \psi \leq \Psi$. Let $\xi \geq 0$ be a constant. Define the neighborhood $\mathcal{N}_1(\psi, \Psi, \xi)$ as

$$\mathcal{N}_1(\psi, \Psi, \xi) \equiv \left\{ x \mid \phi(x) \leq \phi^* + \frac{1}{2m} \left( \frac{\Psi \tilde{e}_g + \xi}{\psi} \right)^2 \right\}$$  \hfill (57)

and the gradient noise to signal ratio $\delta(x)$ as

$$\delta(x) := \frac{\|e(x)\|_2}{\|\nabla \phi(x)\|_2}.$$  \hfill (58)

For all $x \notin \mathcal{N}_1(\psi, \Psi, \xi)$, it is true that

$$\nabla \phi(x)^T H g(x) > \xi \|\nabla \phi(x)\|_2 > \frac{\Psi \tilde{e}_g + \xi}{\psi}$$  \hfill (59)

and $\delta(x) < \frac{\psi \tilde{e}_g}{\Psi \tilde{e}_g + \xi}$. Contrapositively, for all $x$ such that $\delta(x) \geq \frac{\psi \tilde{e}_g}{\Psi \tilde{e}_g + \xi}$ or $\nabla \phi(x)^T H g(x) \leq \xi \|\nabla \phi(x)\|_2$, it is true that $x \in \mathcal{N}_1(\psi, \Psi, \xi)$.

**Proof** See Appendix 5.

**Lemma 3** (Conditions For $\nabla \phi(x)^T H g(x) \geq \varepsilon \nabla \phi(x)^T g(x)$ To Hold) Suppose the conditions required for Lemma 2 to hold apply. Let $A > 0$ be a constant. Set $\xi = A \Psi \tilde{e}_g > 0$ in Lemma 2. For all $x \notin \mathcal{N}_1(\psi, \Psi, A \Psi \tilde{e}_g)$, where

$$\mathcal{N}_1(\psi, \Psi, A \Psi \tilde{e}_g) \equiv \left\{ x \mid \phi(x) \leq \phi^* + \frac{1}{2m} \left( \frac{(1 + A) \Psi \tilde{e}_g}{\psi} \right)^2 \right\},$$  \hfill (60)

the inequality $\nabla \phi(x)^T H g(x) \geq \varepsilon \nabla \phi(x)^T g(x)$ is guaranteed to hold for

$$0 < \varepsilon < \frac{A \Psi \Psi}{((1 + A) \Psi + \psi)}.$$  \hfill (61)

**Proof** See Appendix 6.

Lemmas 2 and 3 allow us to establish the following convergence properties in the context of SP-BFGS updating. First, we use Lemmas 2 and 3 to prove Theorem 4. Theorem 4 establishes that if one chooses $\beta_k$ such that the bounds $\psi I \leq H_k \leq \Psi I$ hold for
all \( k \) (i.e. the eigenvalues of the approximate inverse Hessian are uniformly bounded from above and below for all \( k \)), under additional conditions a worst case analysis shows that an approach using a sufficiently small fixed step size \( \alpha \) approaches \( \mathcal{N}_1(\psi, \Psi, A\Psi \tilde{e}_g) \) at a linear rate as \( k \to \infty \). Second, we use Lemma 2, Lemma 3, and Theorem 4 to prove Theorem 5. Theorem 5 establishes a linear convergence rate for a backtracking line search approach in the presence of noisy function and gradient evaluations.

### 5.2.1 Fixed step size analysis

For a general quasi-Newton iteration of the form

\[
x_{k+1} = x_k - \alpha H_k g_k
\]

with fixed step size \( \alpha \) and \( H_k > 0 \), Theorem 4 establishes linear convergence to the region \( \mathcal{N}_1(\psi, \Psi, A\Psi \tilde{e}_g) \) from Lemma 3.

**Theorem 4** (Linear Convergence For Sufficiently Small Fixed \( \alpha \)) Suppose the conditions required for Lemmas 2 and 3 to hold apply. Let \( \{x_k\} \) be the iterates generated by (62), where the fixed step size \( \alpha \) satisfies

\[
0 < \alpha \leq \frac{\varepsilon}{M^2\Psi^2} < \frac{\Psi}{M^2 \left( \frac{A}{(1 + A)\Psi + \psi} \right)}.
\]

Then for all \( k \) such that \( x_k \not\in \mathcal{N}_1(\psi, \Psi, A\Psi \tilde{e}_g) \), one has the Q-linear convergence result

\[
\phi_{k+1} - \left[ \phi^* + \frac{1}{2m} \left( \frac{(1 + A)\Psi \tilde{e}_g}{\psi} \right)^2 \right] \leq (1 - \alpha m) \left( \phi_k - \left[ \phi^* + \frac{1}{2m} \left( \frac{(1 + A)\Psi \tilde{e}_g}{\psi} \right)^2 \right] \right).
\]

Similarly, for any \( x_0 \not\in \mathcal{N}_1(\psi, \Psi, A\Psi \tilde{e}_g) \), one has the R-linear convergence result

\[
\phi_k - \phi^* \leq (1 - \alpha m) \left( \phi_0 - \left[ \phi^* + \frac{1}{2m} \left( \frac{(1 + A)\Psi \tilde{e}_g}{\psi} \right)^2 \right] \right) + \frac{1}{2m} \left( \frac{(1 + A)\Psi \tilde{e}_g}{\psi} \right)^2.
\]

**Proof** See Appendix 7. \( \square \)

Theorem 4 above can be considered a type of quasi-Newton extension of Theorem 4.1 from [2], with Theorem 4.1 from [2] laying the foundation for Theorem 4.2 from [2]. Similarly, Theorem 4 lays the foundation for Theorem 5 below.

### 5.2.2 Backtracking line search analysis

We now consider a general quasi-Newton iteration of the form

\[
x_{k+1} = x_k - \alpha_k H_k g_k
\]

where the step size \( \alpha_k \) is computed by a backtracking line search. The backtracking line search is based on the relaxed Armijo condition (38), and has backtracking factor \( 0 < \tau < 1 \) and initial step size \( \alpha^0 \). If a trial value \( \alpha_k \) does not satisfy the relaxed
Armijo condition, backtracking sets the new value to a fixed fraction $\tau$ of the failed trial value (i.e. $\alpha_k \leftarrow \tau \alpha_k$). Theorem 5 establishes linear convergence toward a region $\mathcal{N}_2(\bar{\eta})$ where noise in function evaluations prevents further progress guarantees

$$ \mathcal{N}_2(\bar{\eta}) \equiv \left\{ x \mid \phi(x) \leq \phi^* + \bar{\eta} \right\} \quad (66) $$

where $\bar{\eta}$ is given in (69).

**Theorem 5** (Linear Convergence For Backtracking Line Search $\alpha_k$) Suppose the conditions required for Lemmas 2 and 3 to hold apply. Further suppose that Assumption 1 holds as well. Let $\{x_k\}$ be the iterates generated by (65), where the step size $\alpha_k$ is the maximum value in $\{\tau^j \alpha^0 \mid j = 0, 1, 2, \ldots\}$ satisfying the relaxed Armijo condition (38) with $\bar{\epsilon}_f < \epsilon_A$ and

$$ 0 < c_1 \leq \frac{\varepsilon}{2 \Psi} \left( 1 - \frac{\psi}{(1 + A\Psi)} \right) < \frac{\Psi A ((1 + A\Psi) - \psi)}{2 (1 + A\Psi + \psi)^2}. \quad (67) $$

Then, for all $k$ such that $x_k \notin \mathcal{N}_1(\psi, \Psi, A\Psi \bar{\epsilon}_f, A\Psi \bar{\epsilon}_f) \cup \mathcal{N}_2(\bar{\eta})$, one has the $Q$-linear convergence result

$$ \phi_{k+1} - [\phi^* + \bar{\eta}] \leq \rho(\phi_k - [\phi^* + \bar{\eta}]) \quad (68) $$

where

$$ \rho = \left( 1 - \frac{2 mc_1 \tau \varepsilon (1 - \frac{\psi}{(1 + A\Psi)^2})}{\Psi M} \right)^2, \quad \bar{\eta} = \frac{\Psi M}{mc_1 \tau \varepsilon (1 - \frac{\psi}{(1 + A\Psi)^2})^2} (\epsilon_A + \bar{\epsilon}_f). \quad (69) $$

Define

$$ K := \min_k \{ k \in \mathbb{N} \mid x_k \in \mathcal{N}_1(\psi, \Psi, A\Psi \bar{\epsilon}_f) \} \quad (70) $$

as the index of the first iterate to enter $\mathcal{N}_1(\psi, \Psi, A\Psi \bar{\epsilon}_f) \cup \mathcal{N}_2(\bar{\eta})$ (set $K = +\infty$ if no such iterate exists). Similarly, for any $x_0 \notin \mathcal{N}_1(\psi, \Psi, A\Psi \bar{\epsilon}_f) \cup \mathcal{N}_2(\bar{\eta})$, one has the $R$-linear convergence result

$$ \phi_k - \phi^* \leq \rho^k(\phi_0 - [\phi^* + \bar{\eta}]) + \bar{\eta}, \quad \forall k \leq K. \quad (71) $$

**Proof** See Appendix 8. \qed

Theorem 5 shows that if one chooses $\beta_k$ such that $\psi I \leq H_k \leq \Psi I$ holds for all $k$, under additional conditions it is possible to guarantee SP-BFGS iterations converge linearly towards the region where function noise can dominate (i.e. $\mathcal{N}_2(\bar{\eta})$), with the linear progress rate no longer guaranteed once the iterates reach...
the region where gradient noise can dominate (i.e. \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \)). Note that because both \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \) and \( \mathcal{N}_2(\bar{\eta}) \) are defined in terms of the level sets of \( \phi \), \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \) contains \( \mathcal{N}_2(\bar{\eta}) \) and/or \( \mathcal{N}_2(\bar{\eta}) \) contains \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \) depending on the values of the relevant parameters. Mathematically,

\[
\mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \subseteq \mathcal{N}_2(\bar{\eta}) \tag{72}
\]

and/or

\[
\mathcal{N}_2(\bar{\eta}) \subseteq \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g). \tag{73}
\]

Having established worst case theoretical performance guarantees in this section, we now investigate the average case performance of SP-BFGS via numerical experiments in the following section.

### 6 Numerical experiments

In this section, we test instances of Algorithm 1 on a diverse set of 33 test problems for unconstrained minimization. The set of test problems includes convex and non-convex functions, and well known pathological functions such as the Rosenbrock function [40] and its relatives. Described in Sect. 6.1, the first test problem is similar to the one used in the numerical experiments section of [45], and involves an ill conditioned quadratic function. The other 32 problems are selected problems from the CUTEst test problem set [16, 19], and are used for tests in Sect. 6.2. Code for running these numerical experiments was written in the Julia programming language [4], and utilizes the NLPModes.jl [38], CUTEst.jl [37], and Distributions.jl [3, 31] packages. In all the numerical experiments that follow, noise \( e(x) \) was added to function evaluations by uniformly sampling from the interval \([-\bar{e}_f, \bar{e}_f]\), and noise \( e(x) \) was added to the gradient evaluations by uniformly sampling from the closed Euclidean ball \( \|x\|_2 \leq \bar{e}_g \).

#### 6.1 Ill conditioned quadratic function with additive gradient noise only

The first test problem is strongly convex and consists of the 4-dimensional quadratic function given by

\[\phi(x) = \frac{1}{2} x^T T x \tag{74}\]

where the eigenvalues of \( T \) are \( \lambda(T) = \{10^{-2}, 1, 10^2, 10^4\} \). Consequently, the strong convexity parameter is \( m = 10^{-2} \), the Lipschitz constant is \( M = 10^4 \), and the condition number of the Hessian \( T \) is \( 10^6 \). For this test problem, no noise was added to the function evaluations (i.e. \( f(x) = \phi(x) \) in (31)), and \( \bar{e}_g = 1 \). As a result, in this scenario the smallest possible \( \mathcal{N}_1(\psi, \Psi, \xi) \) from Lemma 2 can be described as
Following the discussion in Sect. 4.3, we set the penalty parameters via the formula
\[ \beta_k = \frac{\varepsilon}{\|s_k\|_2} + 10^{-10}, \]
which corresponds to a choice of \( N_s = 1 \) in (49). The \( 10^{-10} \) term was added as a small perturbation to provide numerical stability. The step size \( \alpha_k \) was chosen using a backtracking line search based on the sufficient decrease condition (38) with \( p_k = -H_k g_k \), where \( g_k \) is defined by (32), \( \epsilon_A = 0 \), and \( c_1 = 10^{-4} \). At each iteration, backtracking started from the initial step size \( \alpha^0 = 1 \), decreasing by a factor of \( \tau = 0.5 \) each time the sufficient decrease condition failed. If the backtracking line search exceeded the maximum number of 75 backtracks, we set \( \alpha_k = 0 \). However, the maximum number of backtracks was never exceeded when performing experiments with this first test problem.

Algorithm 1 was initialized using the matrix and starting point \( H^0 = I \), \( x^0 = 10^5 \cdot [1, 1, 1, 1]^T \) given in (76), with \( \| \nabla \phi(x^0) \|_2 \approx 10^9 \). Figures 1, 2 and 3 compare the performance of 30 independent runs of SP-BFGS, BFGS, and gradient descent with a line search over a fixed budget of 100 iterations.

For the sake of comparability, BFGS skipped the update if the BFGS curvature condition failed, and SP-BFGS set \( \beta_k = 0 \) if the SP-BFGS curvature condition with the initial choice of \( \beta_k \) failed. The BFGS curvature condition failed an average of 25.7 total iterations per BFGS run, while the SP-BFGS curvature condition with the initial choice of \( \beta_k \) failed an average of 0.6 total iterations per SP-BFGS run. Observe that SP-BFGS reduces the objective function value by several more orders of magnitude compared to BFGS on average, and maintains significantly better inverse Hessian approximations than BFGS in the presence of gradient noise. As expected, gradient descent is not competitive with either SP-BFGS or BFGS.

### 6.2 CUTEst test problems with various additive noise combinations

The remaining 32 test problems were selected from the CUTEst problem set, the successor of CUTEr [17, 18]. At the time of writing, SIF files and descriptions of all 32 test problems can be found at https://www.cuter.rl.ac.uk/Problems/mastsif.shtml. As a brief summary, some of the problems can be interpreted as least squares type problems (e.g. ARGTRIGLS), some of the problems are ill conditioned or singular type problems (e.g. BOXPOWER), some of the problems are well known nonlinear optimization test problems (e.g. ROSENBR) or extensions of them (e.g. ROEN-BRTU, SROENBR), and some of the problems come from real applications (e.g. COATING, HEART6LS, VIBRBEAM). As shown in Tables 2, 3 and 4, the selected CUTEst test problems vary in size from 2-dimensional to 1000-dimensional.

Using these 32 CUTEst test problems and a fixed budget of 2000 objective function evaluations (not 2000 iterations) per test, we tested the performance of SP-BFGS compared to BFGS and gradient descent with various combinations of function and gradient noise levels \( \tilde{\varepsilon}_f \) and \( \tilde{\varepsilon}_g \). For all the experiments in Tables 1, 2
Secant penalized BFGS: a noise robust quasi-Newton method…

and 3, as well as the additional experiments in Appendix 9, BFGS skipped the update if the BFGS curvature condition failed, and SP-BFGS set $\beta_k = 0$ if the SP-BFGS curvature condition with the initial choice of $\beta_k$ failed. In Tables 1, 2 and 6, the SP-BFGS penalty parameter was set as $\beta_k = \frac{10^p}{\varepsilon_k} \| x_k \|_2 + 10^{-10}$, as the authors heuristically discovered setting $N_s = \frac{10^p}{\varepsilon_k}$ works well in practice for a variety of

Fig. 1 Base 10 logarithm of the optimality gap versus the iteration number $k$ for 30 independent runs. After 100 iterations, SP-BFGS has an average $\log_{10}(\phi_{100} - \phi^*)$ of $-5.03$, BFGS has an average $\log_{10}(\phi_{100} - \phi^*)$ of $-1.27$, and gradient descent has an average $\log_{10}(\phi_{100} - \phi^*)$ of $9.77$. Observe that both SP-BFGS and BFGS appear to enter $N_1(1, 1, 0)$, which corresponds to values less than $\log_{10}(50) \approx 1.7$ on the y-axis, but SP-BFGS makes more progress inside $N_1(1, 1, 0)$. Outside of $N_1(1, 1, 0)$, the performance of SP-BFGS and BFGS is almost indistinguishable. Gradient descent never enters $N_1(1, 1, 0)$.
problems. With regards to the backtracking line search based on (38), we set $\alpha^0 = 1$, $\epsilon_A = \epsilon_f$, $c_1 = 10^{-4}$, $\tau = 0.5$, and the maximum number of backtracks as 45. We define $\Delta_{opt} := \log_{10}(\phi_{best} - \phi^*)$ as a measure of the optimality gap, and use $\phi_{best}$ to denote the smallest value of the true function $\phi$ measured at any point
during an algorithm run. The true minimum values \( \phi^* \) for each CUTEst problem were obtained from the SIF file for each CUTEst problem. The sample variance (i.e. the variance with Bessel’s correction) is denoted by \( s^2(\cdot) \).

Table 1 compares the performance of SP-BFGS versus BFGS on the Rosenbrock function (i.e. ROSENBR) corrupted by different combinations of function and gradient noise of varying orders of magnitude. Observe that SP-BFGS outperforms
Table 1 Performance of SP-BFGS versus BFGS on the Rosenbrock function (i.e. ROSENBR) corrupted by noise

| $\tilde{\epsilon}_f$ | $\tilde{\epsilon}_g$ | Mean ($\Delta_{opt}$) | Median ($\Delta_{opt}$) | Min ($\Delta_{opt}$) | Max ($\Delta_{opt}$) | $s^2(\Delta_{opt})$ | Mean(I) |
|----------------------|----------------------|-----------------------|-----------------------|---------------------|---------------------|---------------------|---------|
| **SP-BFGS with no function noise** |
| 0 10^{-4} | -1.4E+01 | -1.4E+01 | -1.8E+01 | -1.2E+01 | +1.4E+00 | 114 |
| 0 10^{-2} | -1.3E+01 | -1.3E+01 | -1.6E+01 | -8.3E+00 | +2.9E+00 | 104 |
| 0 0 | -2.1E+00 | -1.8E+00 | -5.7E+00 | -9.2E-01 | +9.4E-01 | 153 |
| 0 2 | +3.5E-02 | +2.9E-01 | -1.9E+00 | +7.9E-01 | +3.9E-01 | 90 |
| **BFGS with no function noise** |
| 0 10^{-4} | -1.1E+01 | -1.0E+01 | -1.4E+01 | -8.8E+00 | +1.8E+00 | 263 |
| 0 10^{-2} | -6.6E+00 | -6.6E+00 | -9.6E+00 | -4.3E+00 | +1.6E+00 | 281 |
| 0 0 | -1.5E+00 | -1.2E+00 | -3.3E+00 | -5.4E-01 | +6.3E-01 | 279 |
| 0 2 | +1.1E-01 | +4.3E-01 | -2.4E+00 | +6.5E-01 | +4.7E-01 | 373 |
| **SP-BFGS with low function noise level** |
| 10^{-4} 10^{-4} | -1.4E+01 | -1.4E+01 | -1.5E+01 | -1.3E+01 | +1.9E-01 | 1980 |
| 10^{-4} 10^{-2} | -1.0E+01 | -1.0E+01 | -1.2E+01 | -8.0E+00 | +1.3E+00 | 1964 |
| 10^{-4} 0 | -2.1E+00 | -2.0E+00 | -3.6E+00 | -1.6E+00 | +2.0E-01 | 1759 |
| 10^{-4} 2 | +8.7E-02 | +3.1E-01 | -2.2E+00 | +9.1E-01 | +4.5E-01 | 1720 |
| **BFGS with low function noise level** |
| 10^{-4} 10^{-4} | -1.1E+01 | -1.0E+01 | -1.5E+01 | -8.7E+00 | +1.7E+00 | 1980 |
| 10^{-4} 10^{-2} | -6.6E+00 | -6.5E+00 | -8.8E+00 | -4.7E+00 | +1.2E+00 | 1975 |
| 10^{-4} 0 | -1.2E+00 | -1.1E+00 | -1.8E+00 | -8.6E-01 | +5.9E-02 | 1936 |
| 10^{-4} 2 | +9.5E-02 | +5.1E-01 | -3.1E+00 | +9.2E-01 | +8.5E-01 | 1934 |
| **SP-BFGS with medium function noise level** |
| 10^{-2} 10^{-4} | -1.4E+01 | -1.4E+01 | -1.5E+01 | -1.3E+01 | +3.4E-01 | 1981 |
| 10^{-2} 10^{-2} | -1.0E+01 | -1.0E+01 | -1.3E+01 | -7.5E+00 | +1.5E+00 | 1977 |
| 10^{-2} 0 | -3.4E+00 | -3.0E+00 | -7.5E+00 | -2.0E+00 | +1.7E+00 | 1934 |
| 10^{-2} 2 | -1.8E-01 | +1.7E-01 | -3.7E+00 | +7.4E-01 | +1.0E+00 | 1890 |
| **BFGS with medium function noise level** |
| 10^{-2} 10^{-4} | -1.1E+01 | -1.1E+01 | -1.5E+01 | -8.5E+00 | +1.4E+00 | 1981 |
| 10^{-2} 10^{-2} | -6.7E+00 | -6.7E+00 | -1.0E+01 | -4.9E+00 | +1.7E+00 | 1979 |
| 10^{-2} 0 | -1.8E+00 | -1.5E+00 | -3.8E+00 | -9.1E-01 | +6.3E-01 | 1961 |
| 10^{-2} 2 | +1.4E-01 | +3.9E-01 | -2.3E+00 | +8.5E-01 | +6.1E-01 | 1953 |
| **SP-BFGS with high function noise level** |
| 10 10^{-4} | -1.4E+01 | -1.4E+01 | -1.5E+01 | -1.3E+01 | +2.2E-01 | 1980 |
| 10 10^{-2} | -1.0E+01 | -1.0E+01 | -1.2E+01 | -7.3E+00 | +9.6E-01 | 1978 |
| 10 10 | -3.1E+00 | -2.8E+00 | -5.1E+00 | -1.7E+00 | +8.9E-01 | 1969 |
| 10 2 | -2.2E-01 | +1.1E-02 | -1.9E+00 | +8.4E-01 | +7.6E-01 | 1943 |
| **BFGS with high function noise level** |
| 10 10^{-4} | -1.1E+01 | -1.1E+01 | -1.3E+01 | -9.0E+00 | +1.4E+00 | 1980 |
| 10 10^{-2} | -6.7E+00 | -6.4E+00 | -9.1E+00 | -5.0E+00 | +1.5E+00 | 1980 |
| 10 10 | -1.8E+00 | -1.4E+00 | -5.3E+00 | -8.2E-01 | +1.1E+00 | 1973 |
| 10 2 | -2.9E-02 | +3.7E-01 | -2.1E+00 | +8.9E-01 | +7.9E-01 | 1965 |
Secant penalized BFGS: a noise robust quasi-Newton method…

BFGS with respect to the mean and median optimality gap for every noise combination in Table 1, sometimes by several orders of magnitude. Table 5 shows the performance of gradient descent in the same noise combination scenarios. Gradient descent outperforms both SP-BFGS and BFGS in $\frac{6}{16} \approx 37\%$ of the noise combination scenarios, while both SP-BFGS and BFGS outperform gradient descent in $\frac{10}{16} \approx 63\%$ of the noise combination scenarios. The behaviour of gradient descent somewhat counterintuitively can improve as the amount of noise in the measured gradient increases. This behaviour can be explained by noting that the Rosenbrock problem is only 2-dimensional, and gradient descent with very noisy gradients can exhibit similar behaviour to random search algorithms, such as variants of implicit filtering [28], which can perform very well on such low dimensional problems.

Tables 2, 3 and 4 compare the performance of SP-BFGS, BFGS, and gradient descent on the 32 CUTEst test problems with both function and gradient noise present. Gradient noise was generated using $\bar{\epsilon}_g = 10^{-4}/\text{uni2016.x}$, and function noise was generated using $\bar{\epsilon}_f = 10^{-4}/\text{uni007C.x}$, both to ensure that noise does not initially dominate function or gradient evaluations. Tables 6, 7 and 8 in Appendix 9 perform the same experiments but with only gradient noise present. Note that as the noise in these numerical experiments is additive, the noise to signal ratio of gradient measurements increases as a stationary point is approached.

Overall, by examining the mean and median columns in Tables 2, 3 and 4, one sees that SP-BFGS outperforms both BFGS and gradient descent on $\frac{19}{32} \approx 59\%$ of the CUTEst problems with both function and gradient noise present, and performs at least as well as the best performing alternative on $\frac{29}{32} \approx 91\%$ of these problems. Equivalently, SP-BFGS was only outperformed by BFGS or gradient descent on $\frac{3}{32} \approx 9\%$ of these problems. Referring to Tables 6, 7 and 8 in Appendix 9, with only gradient noise present, one sees that these percentages become $\frac{18}{32} \approx 56\%$, $\frac{28}{32} \approx 88\%$, and $\frac{4}{32} \approx 12\%$ respectively. Thus, SP-BFGS is clearly the most reliable and best performing algorithm overall.

Although SP-BFGS is clearly the best performing algorithm on average in the numerical experiments, it is worth explaining why it may not always be the best performing algorithm. In brief, if $\beta_k$ is chosen poorly, SP-BFGS may not perform as well as BFGS or gradient descent. In practice, this is most likely to occur for problems where either the curvature is very small and the amount of gradient noise is very large, or the curvature is very large and the amount of gradient noise is very small. Recall that as the standard BFGS update corresponds to the SP-BFGS update in the extreme $\beta_k = +\infty$, the standard BFGS update weights the measured curvature information as heavily as possible. This may be the most effective choice of $\beta_k$ in some situations where the noise is imperceptible compared to the curvature of the objective function. At the other extreme, gradient descent corresponds to SP-BFGS updates in the extreme $\beta_k = 0$, and ignores measured curvature information. Ignoring measured curvature information may be the best strategy in situations where

Table 1 (continued)

$\Delta_{opt} := \log_{10}(\phi_{best} - \phi^*)$ measures the optimality gap, where $\phi_{best}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. The number of objective function evaluations is fixed at 2000, but the number of iterations $I$ can vary. Statistics are calculated from a sample of 30 runs per algorithm.
the true curvature of the objective function is imperceptible compared to noise.

Although we have provided (49) and (50) as guides for choosing $\beta_k$, there is certainly more research to be done regarding how to best choose $\beta_k$.

Table 2: Performance of SP-BFGS on 32 selected CUTEst test problems with noise added to both function and gradient evaluations

| Problem      | Dim. | Mean ($\Delta_{opt}$) | Median ($\Delta_{opt}$) | Min ($\Delta_{opt}$) | Max ($\Delta_{opt}$) | $s^2(\Delta_{opt})$ |
|--------------|------|------------------------|-------------------------|----------------------|----------------------|---------------------|
| SP-BFGS with function and gradient noise |
| ARGTRIGLS    | 200  | + 4.5E−02              | + 4.8E−02               | + 1.7E−02            | + 8.0E−02            | 2.5E−04             |
| ARWHEAD      | 500  | − 2.5E+00              | − 2.5E+00               | − 2.6E+00            | − 2.5E+00            | 2.6E−04             |
| BEALE        | 2    | − 1.1E+01              | − 1.1E+01               | − 1.4E+01            | − 9.8E+00            | 8.0E−01             |
| BOX3         | 3    | − 7.1E+00              | − 6.8E+00               | − 8.9E+00            | − 6.5E+00            | 6.2E−01             |
| BOXPOWER     | 100  | − 3.8E+00              | − 3.8E+00               | − 4.2E+00            | − 3.5E+00            | 5.0E−02             |
| BROWNBS      | 2    | − 1.2E+00              | − 7.4E+01               | − 5.2E+00            | + 2.0E+00            | 3.5E+00             |
| BROYDNBDLS   | 50   | − 6.2E+00              | − 6.2E+00               | − 6.4E+00            | − 6.0E+00            | 6.9E−03             |
| CHAINWO       | 100  | + 1.7E+00              | + 1.8E+00               | + 7.7E−03            | + 2.1E+00            | 1.6E−01             |
| CHNROSNB     | 50   | − 4.2E+00              | − 4.0E+00               | − 5.5E+00            | − 3.6E+00            | 3.8E−01             |
| COATING     | 134  | − 2.7E−02              | − 1.2E−02               | − 1.3E−01            | + 9.6E−02            | 3.5E−03             |
| COOLHANSLS  | 9    | − 1.2E+00              | − 1.1E+00               | − 1.6E+00            | − 8.7E−01            | 1.7E−02             |
| CUBE         | 2    | − 5.2E+00              | − 4.7E+00               | − 8.9E+00            | − 3.1E+00            | 2.2E+00             |
| CYCLOOCFLS  | 20   | − 8.4E+00              | − 8.5E+00               | − 9.1E+00            | − 6.9E+00            | 3.0E−01             |
| EXTROSNB    | 10   | − 5.2E+00              | − 5.2E+00               | − 5.2E+00            | − 5.1E+00            | 1.3E−03             |
| FMINSRF2    | 64   | − 8.7E+00              | − 8.7E+00               | − 8.7E+00            | − 8.6E+00            | 2.6E−04             |
| GENHUMPS    | 5    | + 4.1E−02              | + 2.4E−01               | − 2.9E+00            | + 7.8E−01            | 4.5E−01             |
| GENROSE     | 5    | − 9.4E+00              | − 9.3E+00               | − 9.9E+00            | − 9.1E+00            | 5.6E−02             |
| HEART6LS   | 6    | − 3.5E−01              | − 2.7E−01               | − 2.0E+00            | + 1.2E+00            | 1.5E+00             |
| HELIX        | 3    | − 6.1E+00              | − 6.0E+00               | − 7.4E+00            | − 4.5E+00            | 5.0E−01             |
| MANCINO    | 30   | − 2.1E+00              | − 2.1E+00               | − 2.5E+00            | − 1.9E+00            | 1.2E−02             |
| METHANB8LS  | 31   | − 3.8E+00              | − 3.9E+00               | − 4.2E+00            | − 3.4E+00            | 3.6E−02             |
| MODBEALE    | 200  | + 1.1E+00              | + 1.0E+00               | + 4.7E−01            | + 1.8E+00            | 1.8E−01             |
| NONDIA      | 10   | − 4.2E−03              | − 4.3E−03               | − 4.4E−03            | − 3.2E−03            | 9.1E−08             |
| POWELLSG   | 4    | − 6.1E+00              | − 6.0E+00               | − 7.9E+00            | − 4.6E+00            | 9.1E−01             |
| POWER       | 10   | − 3.9E+00              | − 3.8E+00               | − 4.9E+00            | − 3.3E+00            | 1.9E−01             |
| ROSENB       | 2    | − 8.6E+00              | − 8.5E+00               | − 1.1E+01            | − 6.3E+00            | 1.8E+00             |
| ROSENBRTU   | 2    | − 1.8E+01              | − 1.8E+01               | − 2.0E+01            | − 1.7E+01            | 4.0E−01             |
| SBRYBND     | 500  | + 3.9E+00              | + 3.9E+00               | + 3.9E+00            | + 3.9E+00            | 9.2E−06             |
| SINEVAL     | 2    | − 1.4E+01              | − 1.4E+01               | − 1.5E+01            | − 1.3E+01            | 3.7E−01             |
| SNAIL       | 2    | − 1.2E+01              | − 1.2E+01               | − 1.4E+01            | − 1.1E+01            | 2.9E−01             |
| SROSENB     | 1000 | + 5.0E−01              | + 5.0E−01               | + 2.9E−01            | + 6.8E−01            | 8.6E−03             |
| VIBRBEAM   | 8    | + 1.5E+00              | + 1.5E+00               | + 1.5E+00            | + 2.1E+00            | 2.6E−02             |

The number of objective function evaluations is fixed at 2000. $\Delta_{opt} := \log_{10}(\phi_{best} - \phi^*)$ measures the optimality gap, where $\phi_{best}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. The SP-BFGS penalty parameter was set as $\beta_k = \frac{10^{-8}}{\|x_k\|^2} + 10^{-10}$. For each problem, function noise was generated using $\tilde{\epsilon}_f = 10^{-4}\phi(x^0)$, and gradient noise was generated using $\tilde{\epsilon}_g = 10^{-2}\|\nabla\phi(x^0)\|_2$, where the starting point $x^0$ varies by CUTEst problem.
Table 3 Performance of BFGS on 32 selected CUTEst test problems with noise added to both function and gradient evaluations

| Problem          | Dim. | Mean ($\Delta_{q^p}$) | Median ($\Delta_{q^p}$) | Min ($\Delta_{q^p}$) | Max ($\Delta_{q^p}$) | $\sigma^2(\Delta_{q^p})$ |
|------------------|------|------------------------|-------------------------|----------------------|----------------------|--------------------------|
| BFGS with function and gradient noise |
| ARGTRIGLS        | 200  | + 5.6E−02              | + 5.5E−02               | + 2.4E−02            | + 8.4E−02            | 2.7E−04                  |
| ARWHEAD          | 500  | − 2.5E+00              | − 2.5E+00               | − 2.6E+00            | − 2.5E+00            | 4.1E−04                  |
| BEALE            | 2    | − 7.7E+00              | − 7.8E+00               | − 9.7E+00            | − 6.1E+00            | 7.1E−01                  |
| BOX3             | 3    | − 6.5E+00              | − 6.5E+00               | − 6.7E+00            | − 6.4E+00            | 4.7E−03                  |
| BOXPOWER         | 100  | − 3.7E+00              | − 3.7E+00               | − 4.2E+00            | − 3.4E+00            | 3.4E−02                  |
| BROWNSBS         | 2    | + 6.8E−01              | + 1.3E+00               | − 3.2E+00            | + 3.1E+00            | 2.9E+00                  |
| BROYDNBDLS       | 50   | − 6.0E+00              | − 6.0E+00               | − 6.3E+00            | − 5.7E+00            | 2.6E−02                  |
| CHAINWOO         | 100  | + 1.7E+00              | + 1.7E+00               | + 1.2E+00            | + 2.1E+00            | 5.9E−02                  |
| CHNROSBN         | 50   | − 4.2E+00              | − 4.1E+00               | − 5.7E+00            | − 3.4E+00            | 4.4E−01                  |
| COATING          | 134  | − 3.7E−02              | − 5.7E−02               | − 1.6E−01            | + 8.0E−02            | 4.1E−03                  |
| COOLHANSLS       | 9    | − 1.0E+00              | − 1.0E+00               | − 2.0E+00            | − 4.5E−01            | 7.2E−02                  |
| CUBE             | 2    | − 1.6E+00              | − 1.4E+00               | − 3.6E+00            | − 9.7E−01            | 4.1E−01                  |
| CYCLOOCFLS       | 20   | − 7.2E+00              | − 7.2E+00               | − 9.1E+00            | − 5.8E+00            | 8.7E−01                  |
| EXTROSBN         | 10   | − 5.2E+00              | − 5.2E+00               | − 5.2E+00            | − 5.1E+00            | 1.8E−03                  |
| FMINSRF2         | 64   | − 8.6E+00              | − 8.7E+00               | − 8.8E+00            | − 8.2E+00            | 2.8E−02                  |
| GENHUMPS         | 5    | + 1.2E−01              | + 1.2E−01               | − 1.2E+00            | + 8.1E−01            | 2.3E−01                  |
| GENROSE          | 5    | − 7.5E+00              | − 7.6E+00               | − 9.1E+00            | − 6.2E+00            | 7.3E−01                  |
| HEART6LS         | 6    | + 3.1E−01              | + 6.1E−01               | − 1.9E+00            | + 1.2E+00            | 1.4E+00                  |
| HELIX            | 3    | − 4.5E+00              | − 4.7E+00               | − 7.0E+00            | − 2.7E+00            | 1.1E+00                  |
| MANCINO          | 30   | − 1.6E+00              | − 1.6E+00               | − 1.8E+00            | − 1.3E+00            | 1.3E−02                  |
| METHANBLS        | 31   | − 3.9E+00              | − 3.8E+00               | − 4.4E+00            | − 3.6E+00            | 5.5E−02                  |
| MODBEALE         | 200  | + 1.1E+00              | + 1.1E+00               | + 2.9E−01            | + 1.8E+00            | 1.6E−01                  |
| NONDIA           | 10   | − 3.7E−03              | − 3.8E−03               | − 4.4E−03            | − 2.6E−03            | 3.1E−07                  |
| POWELL SG        | 4    | − 5.2E+00              | − 5.2E+00               | − 7.6E+00            | − 4.2E+00            | 7.1E−01                  |
| POWER            | 10   | − 3.5E+00              | − 3.5E+00               | − 4.1E+00            | − 2.9E+00            | 1.0E−01                  |
| ROSENBR          | 2    | − 5.9E+00              | − 5.5E+00               | − 9.2E+00            | − 4.5E+00            | 1.4E+00                  |
| ROSENBRTU        | 2    | − 1.6E+01              | − 1.6E+01               | − 1.8E+01            | − 1.4E+01            | 1.5E+00                  |
| SBRYBNDS         | 500  | + 3.9E+00              | + 3.9E+00               | + 3.9E+00            | + 3.9E+00            | 2.7E−05                  |
| SINEVAL          | 2    | − 1.1E+01              | − 1.1E+01               | − 1.3E+01            | − 8.9E+00            | 1.3E+00                  |
| SNAIL            | 2    | − 9.4E+00              | − 9.2E+00               | − 1.2E+01            | − 8.0E+00            | 7.2E−01                  |
| SROSENBR         | 1000 | + 5.4E−01              | + 3.4E−01               | + 3.6E−01            | + 7.8E−01            | 6.9E−03                  |
| VIBRBEAM         | 8    | + 1.7E+00              | + 1.7E+00               | + 1.2E+00            | + 2.0E+00            | 2.9E−02                  |

The number of objective function evaluations is fixed at 2000. $\Delta_{q^p} := \log_{10}(\phi_{best} - \phi^*)$ measures the optimality gap, where $\phi_{best}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. For each problem, function noise was generated using $\tilde{\epsilon}_f = 10^{-4}[\phi(x^0)]$, and gradient noise was generated using $\tilde{\epsilon}_g = 10^{-4}\|\nabla\phi(x^0)\|_2$, where the starting point $x^0$ varies by CUT-Est problem.
Table 4 Performance of gradient descent on 32 selected CUTEst test problems with noise added to both function and gradient evaluations

| Problem          | Dim. | Mean ($\Delta_{o\phi}$) | Median ($\Delta_{o\phi}$) | Min ($\Delta_{o\phi}$) | Max ($\Delta_{o\phi}$) | $s^2(\Delta_{o\phi})$ |
|------------------|------|--------------------------|---------------------------|------------------------|------------------------|---------------------|
| **Gradient descent with function and gradient noise** |      |                          |                           |                        |                        |                     |
| ARGTRIGLS        | 200  | -2.2E+00                 | -2.2E+00                  | -2.2E+00               | -2.2E+00               | 5.6E-04             |
| ARWHEAD          | 500  | -1.4E+00                 | -1.4E+00                  | -1.6E+00               | -1.3E+00               | 4.6E-03             |
| BEALE            | 2    | -4.1E+00                 | -4.1E+00                  | -4.4E+00               | -3.8E+00               | 2.1E-02             |
| BOX3             | 3    | -4.5E+00                 | -4.5E+00                  | -4.7E+00               | -4.4E+00               | 3.4E-03             |
| BOXPOWER         | 100  | -9.4E-01                 | -9.3E-01                  | -9.9E-01               | -8.9E-01               | 5.1E-04             |
| BROWNBS          | 2    | +1.2E+01                 | +1.2E+01                  | +1.2E+01               | +1.2E+01               | 7.4E-01             |
| BROYDNBDLS       | 50   | -2.4E+00                 | -2.4E+00                  | -2.8E+00               | -2.1E+00               | 2.5E-02             |
| CHAINWOO         | 100  | +2.6E+00                 | +2.6E+00                  | +2.6E+00               | +2.6E+00               | 1.4E-06             |
| CHNROSNB         | 50   | +1.6E+00                 | +1.6E+00                  | +1.6E+00               | +1.6E+00               | 9.9E-06             |
| COATING          | 134  | +2.6E+00                 | +2.6E+00                  | +2.6E+00               | +2.7E+00               | 2.3E-03             |
| COOLHANSLS       | 9    | +1.1E+00                 | +1.1E+00                  | +9.1E-01               | +1.3E+00               | 5.9E-03             |
| CUBE             | 2    | -9.2E-01                 | -9.6E-01                  | -1.0E+00               | -3.8E-01               | 1.9E-02             |
| CYCLOOFLS        | 20   | -2.5E+00                 | -2.7E+00                  | -3.5E+00               | -2.2E-01               | 9.6E-01             |
| EXTROSNB         | 10   | -1.7E+00                 | -1.7E+00                  | -1.8E+00               | -1.6E+00               | 1.5E-03             |
| FMINSRF2         | 64   | -1.4E+00                 | -1.4E+00                  | -1.4E+00               | -1.4E+00               | 5.6E-06             |
| GENHUMPS         | 5    | +6.2E-01                 | +6.6E-01                  | +3.9E-02               | +9.1E-01               | 3.6E-02             |
| GENROSE          | 5    | -1.3E-01                 | -1.3E-01                  | -1.7E-01               | -1.1E-01               | 2.1E-04             |
| HEART6LS         | 6    | +1.5E+00                 | +1.5E+00                  | +1.5E+00               | +1.5E+00               | 1.7E-05             |
| HELIX            | 3    | +9.3E-03                 | +7.9E-03                  | -2.2E-02               | +3.4E-02               | 1.4E-04             |
| MANCINO          | 30   | +3.0E+00                 | +3.1E+00                  | +2.5E+00               | +3.4E+00               | 4.3E-02             |
| METHANBLS        | 31   | -1.7E+00                 | -1.7E+00                  | -1.7E+00               | -1.7E+00               | 5.0E-09             |
| MODBEALE         | 200  | +2.3E+00                 | +2.3E+00                  | +2.3E+00               | +2.3E+00               | 1.0E-05             |
| NONDIA           | 10   | +4.1E-01                 | +4.0E-01                  | +3.9E-01               | +4.2E-01               | 9.0E-05             |
| POWELLSG         | 4    | -2.2E+00                 | -2.2E+00                  | -2.3E+00               | -2.1E+00               | 2.1E-03             |
| POWER            | 10   | -4.0E+00                 | -4.0E+00                  | -4.5E+00               | -3.3E+00               | 8.6E-02             |
| ROSENBR          | 2    | -2.7E+00                 | -2.7E+00                  | -3.3E+00               | -2.2E+00               | 8.9E-02             |
| ROSENBRTU        | 2    | -4.2E-02                 | -2.4E-02                  | -7.1E-02               | -2.0E-02               | 5.6E-04             |
| SBRYBND          | 500  | +3.9E+00                 | +3.9E+00                  | +3.9E+00               | +4.0E+00               | 1.6E-05             |
| SINEVAL          | 2    | +5.5E-01                 | +6.0E-01                  | +2.3E-01               | +6.2E-01               | 1.3E-02             |
| SNAIL            | 2    | -6.3E+00                 | -6.0E+00                  | -8.4E+00               | -4.9E+00               | 1.0E+00             |
| SROSENBR         | 1000 | -4.5E-01                 | -4.5E-01                  | -4.7E-01               | -4.3E-01               | 1.4E-04             |
| VIBRBEAM         | 8    | +3.0E+00                 | +3.0E+00                  | +2.9E+00               | +3.2E+00               | 5.8E-03             |

The number of objective function evaluations is fixed at 2000. $\Delta_{o\phi} := \log_{10}(\phi_{best} - \phi^*)$ measures the optimality gap, where $\phi_{best}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. For each problem, function noise was generated using $\tilde{\epsilon}_f = 10^{-4}||\phi(x^0)||_2$, and gradient noise was generated using $\tilde{\epsilon}_g = 10^{-4}||\nabla \phi(x^0)||_2$, where the starting point $x^0$ varies by CUT-Est problem.
7 Final remarks

In this paper, we introduced SP-BFGS, a new variant of the BFGS method designed to resist the corrupting effects of noise. Motivated by regularized least squares estimation, we derived the SP-BFGS update by applying a penalty method to the secant condition. We argued that with an appropriate choice of penalty parameter, SP-BFGS updating is robust to the corrupting effects of noise that can destroy the performance of BFGS. We empirically validated this claim by performing numerical experiments on a diverse set of over 30 test problems with both function and gradient noise of varying orders of magnitude. The results of these numerical experiments showed that SP-BFGS outperformed both BFGS and gradient descent approximately 59% of the time, and performed at least as good as the best performing alternative approximately 91% of the time. Furthermore, a theoretical analysis confirmed that with appropriate choices of penalty parameter, it is possible to guarantee that SP-BFGS is not corrupted arbitrarily badly by noise, unlike standard BFGS. In the future, we believe it is worth investigating the performance of SP-BFGS in the presence of other types of noise, including multiplicative stochastic noise and deterministic noise, and also believe it is critical to study the use of noise estimation techniques in conjunction with SP-BFGS updating. The authors are also working to publish a limited memory version of SP-BFGS for high dimensional noisy problems.

Appendix 1: Proof of Theorem 1

To produce the SP-BFGS update, we first rearrange (26a), revealing that

$$ (H - H_k) = -W^{-1}(uy_k^T + \Gamma^T - \Gamma)W^{-1} $$

(77)

and so the symmetry requirement that $H = H^T$ means transposing (77) gives

$$ uy_k^T + \Gamma^T - \Gamma = (uy_k^T + \Gamma^T - \Gamma)^T = y_ku^T + \Gamma - \Gamma^T $$

(78)

which rearranges to

$$ \Gamma^T - \Gamma = \frac{1}{2}(y_ku^T - uy_k^T) $$

(79)

and so

$$ (H - H_k) = -\frac{1}{2}W^{-1}(y_ku^T + uy_k^T)W^{-1}. $$

(80)

Next, we right multiply (80) by $y_k$ to get

$$ (H - H_k)y_k = -\frac{1}{2}W^{-1}\left(y_ku^TW^{-1}y_k + u(y_k^TW^{-1}y_k)\right) $$

(81)

and use (26b) to get that
\[ s_k + \frac{W^{-1}u}{\beta_k} - H_k y_k = -\frac{1}{2} W^{-1} \left( y_k u^T W^{-1} y_k + u(y_k^T W^{-1} y_k) \right). \] (82)

We now left multiply both sides by \(-2W\) and rearrange, giving

\[ -2W(s_k - H_k y_k) = y_k u^T W^{-1} y_k + u \left( y_k^T W^{-1} y_k + \frac{2}{\beta_k} \right). \] (83)

This can be rearranged so that \(u\) is isolated, giving

\[
\begin{align*}
u &= \frac{-2W(s_k - H_k y_k) - y_k u^T W^{-1} y_k}{y_k^T W^{-1} y_k + \frac{2}{\beta_k}} \\
&= \frac{-2W(s_k - H_k y_k) + y_k u^T W^{-1} y_k}{y_k^T W^{-1} y_k + \frac{2}{\beta_k}}.
\end{align*}
\] (84)

To get rid of the \(u^T\) on the right hand side, we first left multiply both sides by \(y_k^T W^{-1}\), and then transpose to get

\[
u^T W^{-1} y_k = \frac{2(s_k - H_k y_k)^T y_k + (y_k^T W^{-1} y_k)(u^T W^{-1} y_k)}{y_k^T W^{-1} y_k + \frac{2}{\beta_k}}.
\] (85)

where we have taken advantage of the fact that the transpose of a scalar returns the same scalar. This now allows us to solve for \(u^T W^{-1} y_k\) using some basic algebra, and resulting in

\[
u^T W^{-1} y_k = \frac{(s_k - H_k y_k)^T y_k}{y_k^T W^{-1} y_k + \frac{1}{\beta_k}}.
\] (86)

Substituting (86) into (84) gives

\[
u = \frac{y_k y_k^T (s_k - H_k y_k)}{(y_k^T W^{-1} y_k + \frac{2}{\beta_k})(y_k^T W^{-1} y_k + \frac{1}{\beta_k})} - \frac{2W(s_k - H_k y_k)}{y_k^T W^{-1} y_k + \frac{2}{\beta_k}}.
\] (87)

Now, if we substitute the expression for \(u\) in (87) into (80), after some simplification we get

\[
(H - H_k) = \frac{1}{(y_k^T W^{-1} y_k + \frac{2}{\beta_k})} \left( (s_k - H_k y_k)^T W^{-1} + W^{-1} y_k (s_k - H_k y_k)^T - \frac{y_k^T (s_k - H_k y_k)}{y_k^T W^{-1} y_k + \frac{2}{\beta_k}} \right) W^{-1} y_k^T.
\]

Now, we further simplify by applying that \(W s_k = y_k\), and thus \(W^{-1} y_k = s_k\), revealing

\[
H = H_k + \frac{(s_k - H_k y_k) s_k^T + s_k (s_k - H_k y_k)^T}{(y_k^T s_k + \frac{2}{\beta_k})} - \frac{y_k^T (s_k - H_k y_k)}{(y_k^T s_k + \frac{2}{\beta_k})} \frac{y_k^T s_k + \frac{1}{\beta_k}}{s_k^T s_k}.
\] (88)
which, after a bit of algebra, reveals that the update formula solving the system defined by (26a), (26b), and (26c) can be expressed as

$$ H^* = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k^T}{(y_k^T s_k + \frac{2}{\beta_k})} \left[ \frac{y_k^T s_k + \frac{2}{\beta_k}}{(y_k^T s_k + \frac{2}{\beta_k})(y_k^T s_k + \frac{1}{\beta_k})} \right] s_k s_k^T. \tag{89} $$

We can make (89) look similar to the common form of the BFGS update given in (19) by defining the two quantities $\gamma_k$ and $\omega_k$ as in (28) and observing that completing the square gives

$$ H^* = \left(I - \frac{\gamma_k}{\omega_k} s_k y_k^T\right)H_k \left(I - \frac{\gamma_k}{\omega_k} y_k s_k^T\right) + \omega_k \left[ \frac{y_k}{\omega_k} + (y_k - \omega_k) y_k^T H_k y_k \right] s_k s_k^T. \tag{90} $$

which is equivalent to

$$ H^* = \left(I - \omega_k y_k s_k^T\right)H_k \left(I - \omega_k y_k s_k^T\right) + \omega_k \left[ \frac{y_k}{\omega_k} + (y_k - \omega_k) y_k^T H_k y_k \right] s_k s_k^T. \tag{91} $$

concluding the proof.

**Appendix 2: Proof of Lemma 1**

The $H_{k+1}$ given by (27) has the general form

$$ H_{k+1} = G^T H_k G + d_k s_k^T \tag{92} $$

with the specific choices

$$ G = I - \omega_k y_k s_k^T, \quad d = \omega_k \left[ \frac{y_k}{\omega_k} + (y_k - \omega_k) y_k^T H_k y_k \right]. \tag{93} $$

By definition, $H_{k+1}$ is positive definite if

$$ v^T H_{k+1} v > 0, \quad \forall v \in \mathbb{R}^n \setminus 0. \tag{94} $$

We first show that (29) is a sufficient condition for $H_{k+1}$ to be positive definite, given that $H_k$ is positive definite. By applying (92) to (94), we see that

$$ v^T \left(G^T H_k G + d_k s_k^T\right) v > 0, \quad \forall v \in \mathbb{R}^n \setminus 0 \tag{95} $$

must be true for the choices of $G$ and $d$ in (93) if $H_{k+1}$ is positive definite. Substituting (93) into (95) reveals that

$$ \left(v - \omega_k (s_k^T v) y_k\right)^T H_k \left(v - \omega_k (s_k^T v) y_k\right) + \omega_k \left[ \frac{y_k}{\omega_k} + (y_k - \omega_k) y_k^T H_k y_k \right] (s_k^T v)^2 > 0. \tag{96} $$
must be true for all \( v \in \mathbb{R}^n \setminus 0 \) if \( H_{k+1} \) is positive definite. Both \( (s_k^T v)^2 \) and \( v^T G^T H_k G v \) are always nonnegative. To see that \( v^T G^T H_k G v \geq 0 \), note that because \( H_k \) is positive definite, it has a principal square root \( H_k^{1/2} \), and so

\[
v^T G^T H_k G v = v^T G^T H_k^{1/2} H_k^{1/2} G v = \left\| H_k^{1/2} G v \right\|^2_2 \geq 0. \tag{97}
\]

We now observe that if \( d > 0 \), the right term \( d(s_k^T v)^2 \) in (96) is zero if and only if \( (s_k^T v) = 0 \). However, if \( (s_k^T v) = 0 \), then the left term \( v^T G^T H_k G v \) in (96) is zero only when \( v = 0 \). Hence, the condition \( d > 0 \) guarantees that (96) is true for all \( v \) excluding the zero vector, and thus that \( H_{k+1} \) is positive definite. The condition \( d > 0 \) expands to

\[
\gamma_k + \omega_k (\gamma_k - \omega_k) y_k^T H_k y_k > 0. \tag{98}
\]

Using the definitions of \( \gamma_k \) and \( \omega_k \) in (28), it is clear that \( (\gamma_k - \omega_k) \geq 0 \), as \( \beta_k \) can only take nonnegative values. Furthermore, as \( H_k \) is positive definite, \( y_k^T H_k y_k \geq 0 \) for all \( y_k \). As it is possible for \( (\gamma_k - \omega_k) y_k^T H_k y_k \) to be zero, we require \( \gamma_k > 0 \). The condition \( \gamma_k > 0 \) immediately gives (29), as \( \gamma_k \) can only be positive if the denominator in its definition is positive. Finally, as \( \beta_k \) can only take nonnegative values, (29) also ensures that \( \omega_k \) is nonnegative, and so when (29) is true, \( \omega_k (\gamma_k - \omega_k) y_k^T H_k y_k \geq 0 \). In summary, we have shown that the condition (29) ensures that the left term in (98) is positive, and the right term nonnegative, so \( d > 0 \), and thus \( H_{k+1} \) is positive definite.

We now show that (29) is a necessary condition for \( H_{k+1} \) to be positive definite, given that \( H_k \) is positive definite. If \( H_{k+1} \) is positive definite, then

\[
y_k^T H_{k+1} y_k > 0 \tag{99}
\]

assuming \( y_k \neq 0 \). Substituting (26b) into (99) gives

\[
y_k^T \left[ s_k + \frac{W^{-1} u}{\beta_k} \right] > 0 \tag{100}
\]

and using (86) shows that (100) is equivalent to

\[
y_k^T \left[ s_k + \frac{\gamma_k (H_k y_k - s_k)}{\beta_k} \right] > 0. \tag{101}
\]

Now, some algebra shows that
and we also know that because $H_k$ is positive definite, $y_k^T H_k y_k > 0$ for all $y_k \neq 0$, by definition $\beta_k \geq 0$, and by the definition of the square of a real number, $(y_k^T s_k)^2 \geq 0$. As a result,

$$y_k^T \left[ s_k + \frac{y_k^T (H_k y_k - s_k)}{\beta_k} \right] = y_k^T s_k + \frac{1}{1 + \beta_k y_k^T s_k} \left[ y_k^T H_k y_k - y_k^T s_k \right]$$

$$= \left( 1 - \frac{1}{1 + \beta_k y_k^T s_k} \right) y_k^T s_k + \left( \frac{1}{1 + \beta_k y_k^T s_k} \right) y_k^T H_k y_k$$

$$= \left( \frac{\beta_k y_k^T s_k}{1 + \beta_k y_k^T s_k} \right) y_k^T s_k + \left( \frac{1}{1 + \beta_k y_k^T s_k} \right) y_k^T H_k y_k$$

$$= \frac{\beta_k (y_k^T s_k)^2 + y_k^T H_k y_k}{1 + \beta_k y_k^T s_k}$$

(102)

is guaranteed only if the denominator $1 + \beta_k y_k^T s_k$ is positive, which occurs when

$$s_k^T y_k > -\frac{1}{\beta_k}.$$  (104)

This establishes that (29) is a necessary condition for $H_{k+1}$ to be positive definite, given that $H_k$ is positive definite, and concludes the proof.

**Appendix 3: Proof of Theorem 2**

The Sherman-Morrison-Woodbury formula says

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} + VA^{-1} U)^{-1} VA^{-1}.$$  (105)

Now, observe that the SP-BFGS update (27) can be written in the factored form

$$H_{k+1} = H_k + \omega_k \left[ s_k \ H_k y_k \right] \left[ y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) - 1 \right] \left[ s_k^T \ y_k^T H_k \right].$$  (106)

Applying the Sherman-Morrison-Woodbury formula (105) to the factored SP-BFGS update (106) with
\[ A = H_k, \]
\[ U = \omega_k \begin{bmatrix} s_k & H_k y_k \end{bmatrix}, \]
\[ C = \begin{bmatrix} \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) - 1 & 0 \\ 0 & 0 \end{bmatrix}, \]
\[ V = \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix} \]
yields
\[ H_{k+1}^{-1} = H_k^{-1} - H_k^{-1} \omega_k \begin{bmatrix} s_k & H_k y_k \end{bmatrix} \left( C^{-1} + VH_k^{-1} U \right)^{-1} \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix} H_k^{-1}. \]

Inverting \( C \) here gives
\[ C^{-1} = \begin{bmatrix} \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) - 1 & 0 \\ 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & -1 \\ -1 & \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \end{bmatrix} \]
and we also have
\[ VH_k^{-1} U = \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix} H_k^{-1} \omega_k \begin{bmatrix} s_k & H_k y_k \end{bmatrix} \]
\[ = \omega_k \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix} \begin{bmatrix} H_k^{-1} s_k & y_k \\ H_k^{-1} y_k & s_k \end{bmatrix} \]
\[ = \begin{bmatrix} \omega_k s_k^T H_k^{-1} s_k & \omega_k s_k^T y_k \\ \omega_k y_k^T s_k & \omega_k y_k^T y_k \end{bmatrix} \]
which is just a \( 2 \times 2 \) matrix with real entries. Now, it becomes clear that
\[ (C^{-1} + VH_k^{-1} U) = \left( \begin{bmatrix} \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) - 1 & 0 \\ 0 & 0 \end{bmatrix}^{-1} + \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix} H_k^{-1} \omega_k \begin{bmatrix} s_k & H_k y_k \end{bmatrix} \right) \]
\[ = \begin{bmatrix} \omega_k s_k^T H_k^{-1} s_k & 1 + \omega_k s_k^T y_k \\ 1 + \omega_k y_k^T s_k & \omega_k y_k^T y_k - \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \end{bmatrix}. \]

For notational compactness, let
\[ D = (C^{-1} + VH_k^{-1} U) \]
\[ = \begin{bmatrix} \omega_k s_k^T H_k^{-1} s_k & 1 + \omega_k s_k^T y_k \\ 1 + \omega_k y_k^T s_k & \omega_k y_k^T y_k - \gamma_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \end{bmatrix} \]
so
\[ D^{-1} = \frac{1}{\det(D)} \begin{bmatrix} \omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) & 1 - \omega_k s_k^T y_k \\ 1 - \omega_k y_k^T s_k & \omega_k s_k^T H_k^{-1} s_k \end{bmatrix} \]

where the determinant of \( D \) is

\[
\det(D) = \left( \omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \right) \left( \omega_k s_k^T H_k^{-1} s_k - (1 - \omega_k y_k^T s_k)^2 \right)
\]

and we have used the fact that \( y_k^T s_k = s_k^T y_k \), as this is a scalar quantity. Next,

\[
U \det(D) D^{-1} V = U \begin{bmatrix} \omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) & 1 - \omega_k s_k^T y_k \\ 1 - \omega_k y_k^T s_k & \omega_k s_k^T H_k^{-1} s_k \end{bmatrix} \begin{bmatrix} s_k^T \\ y_k^T H_k \end{bmatrix}
\]

so \( U \det(D) D^{-1} V \) fully expanded becomes

\[
U \det(D) D^{-1} V = \omega_k \left[ (\omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right)) u_1 + (1 - \omega_k y_k^T s_k)^2 + (1 - \omega_k y_k^T s_k)^2 (s_k^T y_k H_k)^2 \right].
\]

This looks rather ugly at the moment, but we continue by breaking the problem down further, noting that

\[
\alpha_k \left( \omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \right) \neq \left( \omega_k y_k^T H_k y_k - y_k \left( \frac{1}{\omega_k} + y_k^T H_k y_k \right) \right) + \left( (1 - \omega_k y_k^T s_k)^2 + (1 - \omega_k y_k^T s_k)^2 (s_k^T y_k H_k)^2 \right).
\]

and

\[
H_j y_k \left( 1 - \omega_k y_k^T s_k \right) s_k^T + \omega_k s_k^T H_k^{-1} s_k y_k^T H_k \right) = \left( 1 - \omega_k y_k^T s_k \right) y_k s_k^T + \omega_k H_j y_k (s_k^T y_k H_k)^2.
\]

The above intermediate results further simplify \( U \det(D) D^{-1} V \) to

\[
U \det(D) D^{-1} V = \omega_k \left[ \left( (1 - \omega_k y_k^T s_k)^2 + (1 - \omega_k y_k^T s_k)^2 (s_k^T y_k H_k)^2 \right) \right].
\]

Left and right multiplying the line immediately above by \( A^{-1} = H_k^{-1} \) gives

\[
H_k^{-1} U \det(D) D^{-1} V y_k = \omega_k \left[ \left( (1 - \omega_k y_k^T s_k)^2 + (1 - \omega_k y_k^T s_k)^2 (s_k^T y_k H_k)^2 \right) \right].
\]

and thus, after dividing out \( \det(D) \) and applying \( B_k = H_k^{-1} \), we arrive at the following final formula.
for the SP-BFGS inverse update, which concludes the proof.

Appendix 4: Proof of Theorem 3

Referring to Theorem 2, taking the trace of both sides of (107) and applying the linearity and cyclic invariance properties of the trace yields

\[
\text{Tr}(B_{k+1}) = \kappa_1 \text{Tr}(B_k) + \kappa_2 \|B_k s_k\|^2 + 2\kappa_3 (y_k^T B_k s_k) + \kappa_4 \|y_k\|^2
\]

(108)

where

\[
\kappa_1 = 1, \quad \kappa_2 = -\frac{\omega_k \hat{D}}{[\hat{D}(\omega_k s_k^T B_k s_k) - (\hat{E})^2]},
\]

(109)

\[
\kappa_3 = -\frac{\omega_k \hat{E}}{[\hat{D}(\omega_k s_k^T B_k s_k) - (\hat{E})^2]}, \quad \kappa_4 = -\frac{(\omega_k)^2 s_k^T B_k s_k}{[\hat{D}(\omega_k s_k^T B_k s_k) - (\hat{E})^2]},
\]

(110)

with \(\hat{D}\) and \(\hat{E}\) defined as

\[
\hat{D} = \left[(\omega_k - \gamma_k)(y_k^T B_k^{-1} y_k) - \frac{y_k^T y_k}{\omega_k}\right], \quad \hat{E} = (1 - \omega_k s_k^T y_k) = \frac{2\omega_k}{\beta_k}
\]

(111)

We now observe that after applying some basic algebra, and recalling that \(B_k\) is positive definite, one can deduce that for all \(\beta_k \in [0, +\infty]\), the following inequalities hold

\[
(\omega_k - \gamma_k) \leq 0, \quad 1 \leq \frac{y_k^T y_k}{\omega_k}, \quad \hat{D} \leq -1, \quad 0 \leq \frac{2\omega_k}{\beta_k} \leq 1.
\]

(112)

By minimizing the absolute value of the common denominator in \(\kappa_2, \kappa_3, \text{ and } \kappa_4\) using the inequalities above, one can obtain the bounds
As a result, and applying \( \lambda_{\max}(B_k) < \text{Tr}(B_k) \) establishes (53). Similarly, referring to (89) reveals the upper bound

\[
\text{Tr}(B_{k+1}) \leq \text{Tr}(B_k) + 2 \alpha_k |y_k^T B_k s_k| + \kappa_4 \|y_k\|^2_2
\]

and applying \( \lambda_{\max}(B_k) < \text{Tr}(B_k) \) establishes (53). Similarly, referring to (89) reveals the upper bound

\[
\text{Tr}(H_{k+1}) \leq \text{Tr}(H_k) + 2 \alpha_k |y_k^T H_k s_k| + \left[ \kappa_k + \alpha_k y_k (y_k^T H_k y_k) \right] \|s_k\|^2_2.
\]

To establish (52), we apply \( \lambda_{\max}(H_k) < \text{Tr}(H_k) \) and \( \alpha_k \leq \gamma_k \) to the line above, and then factor. This completes the proof.

**Appendix 5: Proof of Lemma 2**

As \( \phi \) is \( m \)-strongly convex due to Assumption 3, it is true that

\[
\phi(y) \geq \phi(x) + \nabla \phi(x)^T (y - x) + \frac{m}{2} \|y - x\|^2_2, \quad \forall x, y \in \mathbb{R}^n.
\]

Note that for any fixed \( x \), the right side of (118) provides a global quadratic lower bound on \( \phi \). As these bounds are global lower bounds, minimizing both sides of (118) with respect to \( y \) preserves the inequality, so

\[
\min_y \left\{ \phi(y) \right\} \geq \min_y \left\{ \phi(x) + \nabla \phi(x)^T (y - x) + \frac{m}{2} \|y - x\|^2_2 \right\}
\]

which simplifies to

\[
\phi^* \geq \phi(x) - \frac{1}{2m} \|\nabla \phi(x)\|^2_2.
\]

Proceeding, the inner product condition \( \nabla \phi(x)^T H g(x) > \xi \|\nabla \phi(x)\|_2 \) expands to

\[
\nabla \phi(x)^T H g(x) = \nabla \phi(x)^T H \nabla \phi(x) + \nabla \phi(x)^T H e(x) > \xi \|\nabla \phi(x)\|_2.
\]

The smallest possible value of \( \nabla \phi(x)^T H \nabla \phi(x) \) is

\[
\nabla \phi(x)^T H \nabla \phi(x) \geq \psi \|\nabla \phi(x)\|^2_2.
\]
By applying the Cauchy-Schwarz inequality and Assumption 2, the most negative possible value of $\nabla \phi(x)^T H e(x)$ is

$$\nabla \phi(x)^T H e(x) \geq -\Psi \| \nabla \phi(x) \|_2 \| e(x) \|_2 \geq -\Psi \| \nabla \phi(x) \|_2 \bar{e}_g.$$  

(123)

Thus, we see that if

$$\psi \| \nabla \phi(x) \|_2^2 - \Psi \| \nabla \phi(x) \|_2 \bar{e}_g > \xi \| \nabla \phi(x) \|_2,$$

(124)

which rearranges to

$$\| \nabla \phi(x) \|_2 > \frac{\Psi \bar{e}_g + \xi}{\psi},$$

(125)

then $\nabla \phi(x)^T H g(x) > \xi \| \nabla \phi(x) \|_2$ is guaranteed. Note that (125) implies

$$\nabla \phi(x)^T H g(x) > \xi \| \nabla \phi(x) \|_2 > \xi \left[ \frac{\Psi \bar{e}_g + \xi}{\psi} \right]$$

(126)

when combined with the inner product condition. Combining (125) with Assumption 2 and the definition of the gradient noise to signal ratio $\delta(x)$ given by (58) reveals that

$$\frac{\Psi \bar{e}_g + \xi}{\psi} < \| \nabla \phi(x) \|_2 = \frac{\| e(x) \|_2}{\delta(x)} \leq \frac{\bar{e}_g}{\delta(x)}$$

(127)

and so $\delta(x) < \frac{\psi e_g}{\Psi \bar{e}_g + \xi} \leq \frac{\psi}{\Psi}$

Contrapositively, if $\nabla \phi(x)^T H g(x) \leq \xi \| \nabla \phi(x) \|_2$, then

$$\| \nabla \phi(x) \|_2 \leq \frac{\Psi \bar{e}_g + \xi}{\psi},$$

(128)

or if $\delta(x) \geq \frac{\psi e_g}{\Psi \bar{e}_g + \xi} \geq 0$, then

$$\| \nabla \phi(x) \|_2 \leq \left( \frac{\Psi \bar{e}_g + \xi}{\psi \bar{e}_g} \right) \| e(x) \|_2 \leq \frac{\Psi \bar{e}_g + \xi}{\psi}.$$  

(129)

Squaring either inequality (128) or (129) and then combining it with a rearranged (120) given by

$$\phi(x) - \phi^* \leq \frac{1}{2m} \| \nabla \phi(x) \|_2^2$$

(130)

gives $N_1(\psi, \Psi, \xi)$, completing the proof.

**Appendix 6: Proof of Lemma 3**

Similar to (122) and (123), by using the definition of $\delta(x)$, the lower bound
\[ \nabla \phi(x)^T H g(x) \geq \psi \| \nabla \phi(x) \|_2^2 - \Psi \| \nabla \phi(x) \|_2 \| e(x) \|_2 = (\psi - \Psi \delta(x)) \| \nabla \phi(x) \|_2^2 \]  
(131)

and the upper bound
\[ \varepsilon (1 + \delta(x)) \| \nabla \phi(x) \|_2^2 = \varepsilon \left( \| \nabla \phi(x) \|_2^2 + \| \nabla \phi(x) \|_2 \| e(x) \|_2 \right) \geq \varepsilon \| \nabla \phi(x) \|_2^2 g(x) \]  
(132)
can be established. Observe that if the lower bound (131) is always greater than or equal to the upper bound (132)
\[ (\psi - \Psi \delta(x)) \| \nabla \phi(x) \|_2^2 \geq \varepsilon (1 + \delta(x)) \| \nabla \phi(x) \|_2^2, \]  
(133)
it implies that \( \nabla \phi(x)^T H g(x) \geq \varepsilon \| \nabla \phi(x) \|_2^2 g(x) \). Hence, the condition
\[ \varepsilon \leq \frac{\psi - \Psi \delta(x)}{(1 + \delta(x))} \]  
(134)
implies that \( \nabla \phi(x)^T H g(x) \geq \varepsilon \| \nabla \phi(x) \|_2^2 g(x) \). By applying Lemma 2, we see that for all \( x \notin \mathcal{N}_1(\psi, \Psi, A \Psi \bar{\varepsilon}_g) \), it is true that \( \delta(x) < \frac{\psi}{(1+A)\Psi} \). Thus, setting
\[ \varepsilon < \frac{\psi - \frac{\psi}{(1+A)}}{(1 + \frac{\psi}{(1+A)\Psi})} = \frac{A\psi \Psi}{((1+A)\Psi + \psi)} \]  
(135)
guarantees that \( \nabla \phi(x)^T H g(x) \geq \varepsilon \| \nabla \phi(x) \|_2^2 g(x) \) for all \( x \notin \mathcal{N}_1(\psi, \Psi, A \Psi \bar{\varepsilon}_g) \), completing the proof.

**Appendix 7: Proof of Theorem 4**

As \( \phi \in C^2 \) by Assumption 3, applying Taylor’s theorem and using (62) and strong convexity gives
\[ \phi_{k+1} = \phi_k + \nabla \phi_k^T [x_{k+1} - x_k] + \frac{1}{2} [x_{k+1} - x_k]^T \nabla^2 \phi(u)[x_{k+1} - x_k] \]
\[ \leq \phi_k - a \nabla \phi_k^T H_k g_k + \frac{\alpha^2 M}{2} \| H_k g_k \|_2^2 \]

where \( a \) is some convex combination of \( x_{k+1} \) and \( x_k \). Proceeding, note that the smallest possible region \( \mathcal{N}_1(\psi, \Psi, A \Psi \bar{\varepsilon}_g) \) from Lemma 3 occurs with the choice \( \psi = \Psi \). In this case \( H = \Psi I \), and (59) from Lemma 2 becomes
\[ \nabla \phi_k^T g_k > A \bar{\varepsilon}_g \left[ (1 + A) \bar{\varepsilon}_g \right] > 0 \]  
(136)
and so \( \nabla \phi_k^T g_k > 0 \) if \( x_k \notin \mathcal{N}_1(\psi = \Psi, \Psi, A \Psi \bar{\varepsilon}_g) \). Hence, for all possible choices of \( 0 < \psi \leq \Psi \) in \( \mathcal{N}_1(\psi, \Psi, A \Psi \bar{\varepsilon}_g) \), we have \( \nabla \phi_k^T g_k > 0 \) if \( x_k \notin \mathcal{N}_1(\psi, \Psi, A \Psi \bar{\varepsilon}_g) \). Combining this with Lemma 3 gives
\[
\n\n\text{if } x_k \not\in \mathcal{N}_1(\psi, \Psi, A\Psi \bar{\epsilon}_g). \text{ With (137) in hand, continuing to bound terms gives }
\]
\[
\phi_{k+1} \leq \phi_k - \alpha \nabla \phi_k^T [\nabla \phi_k + e_k] + \frac{\alpha^2 \Psi^2 M}{2} \| \nabla \phi_k + e_k \|^2_2
\]
\[
\leq \phi_k - \alpha \left( \frac{\epsilon}{\psi} - \frac{\alpha \Psi M}{2} \right) \| \nabla \phi_k \|^2_2 - \alpha \left( \frac{\epsilon}{\psi} - \frac{\alpha \Psi M}{2} \right) \| \nabla \phi_k \|_2 \| e_k \|_2 + \frac{\alpha^2 \Psi^2 M}{2} \| e_k \|^2_2
\]
\[
\leq \phi_k - \alpha \left( \frac{\epsilon}{\psi} - \frac{\alpha \Psi M}{2} \right) \| \nabla \phi_k \|^2_2 + \alpha \left( \frac{\epsilon}{\psi} - \frac{\alpha \Psi M}{2} \right) \left[ \frac{1}{2} \| \nabla \phi_k \|^2_2 + \frac{1}{2} \| e_k \|^2_2 \right] + \frac{\alpha^2 \Psi^2 M}{2} \| e_k \|^2_2
\]
\[
\text{where the last inequality follows from expanding}
\]
\[
0 \leq \left( \frac{1}{\sqrt{2}} \| \nabla \phi_k \|_2 - \frac{1}{\sqrt{2}} \| e_k \|_2 \right)^2 = \frac{1}{2} \| \nabla \phi_k \|^2_2 - \| \nabla \phi_k \|_2 \| e_k \|_2 + \frac{1}{2} \| e_k \|^2_2
\]
\[
\text{and using } \alpha \leq \frac{\epsilon}{M\Psi^2} \text{ in (63). Simplifying the last inequality reveals that}
\]
\[
\phi_{k+1} \leq \phi_k - \frac{\alpha \epsilon}{2} \| \nabla \phi_k \|^2_2 + \frac{\alpha \epsilon}{2} \| e_k \|^2_2.
\]
\[
\text{Since } \phi \text{ is } m \text{-strongly convex by Assumption 3, we can apply}
\]
\[
\| \nabla \phi_k \|^2_2 \geq 2m(\phi_k - \phi^*)
\]
\[
\text{which comes from rearranging (120) in the proof of Lemma 2 (see Appendix 5). Combining (140) with (139) and Assumption 2 gives}
\]
\[
\phi_{k+1} \leq \phi_k - \alpha \epsilon m(\phi_k - \phi^*) + \frac{\alpha \epsilon}{2} \left( \frac{(1 + A)\Psi \bar{\epsilon}_g}{\psi} \right)^2.
\]
\[
\text{Subtracting } \phi^* \text{ from both sides, and using the notation } \check{A} := (1 + A), \text{ we get}
\]
\[
\phi_{k+1} - \phi^* \leq (1 - \alpha \epsilon m)(\phi_k - \phi^*) + \frac{\alpha \epsilon}{2} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2
\]
\[
\text{which, by subtracting } \frac{1}{2m} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2 \text{ from both sides and simplifying, gives}
\]
\[
\phi_{k+1} - \phi^* - \frac{1}{2m} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2 \leq (1 - \alpha \epsilon m)(\phi_k - \phi^*) + \frac{\alpha \epsilon}{2} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2 + \frac{1}{2m} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2
\]
\[
= (1 - \alpha \epsilon m)(\phi_k - \phi^*) + (\alpha \epsilon m - 1) \frac{1}{2m} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2
\]
\[
= (1 - \alpha \epsilon m) \left( \phi_k - \left[ \phi^* + \frac{1}{2m} \left( \frac{\check{A} \Psi \bar{\epsilon}_g}{\psi} \right)^2 \right] \right)
\]
\[
\text{thus establishing the Q-linear result. We obtain the R-linear result (64) by recursively applying the worst case bound given by the Q-linear result, noting that in the}
\]
worst case if \( x_0 \not\in \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \), then the sequence of iterates \( \{x_k\} \) remains outside of \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \), only approaching \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \) in the limit \( k \to \infty \).

**Appendix 8: Proof of Theorem 5**

From (139) in Appendix 7, if the step size \( \alpha \leq \frac{\epsilon}{M^2 \epsilon} \) from (63), one has

\[
\phi(x_k - \alpha H_k g_k) \leq \phi(x_k) - \frac{\alpha \epsilon}{2} \left\| \nabla \phi(x_k) \right\|^2 + \frac{\alpha \epsilon}{2} \left\| e(x_k) \right\|^2
\]

(143)

which combines with Assumption 1 to give

\[
f(x_k - \alpha H_k g_k) \leq f(x_k) - \frac{\alpha \epsilon}{2} \left( \left\| \nabla \phi(x_k) \right\|^2 - \left\| e(x_k) \right\|^2 \right) + 2\epsilon_f.
\]

(144)

The relaxed Armijo condition (38) expands to

\[
f(x_k - \alpha H_k g_k) \leq f(x_k) - c_1 \alpha g_k^T H_k g_k + 2\epsilon_A
\]

(145)

and so the strongest possible condition (i.e. the condition requiring the greatest decrease in \( f \)) can be written as

\[
f(x_k - \alpha H_k g_k) \leq f(x_k) - c_1 \alpha \Psi \left\| g_k \right\|^2 + 2\epsilon_A.
\]

(146)

Comparing (144) and (146) reveals that for the bound given by (144) to also imply the bound given by (146), it must be true that

\[-\frac{\alpha \epsilon}{2} \left( \left\| \nabla \phi(x_k) \right\|^2 - \left\| e(x_k) \right\|^2 \right) + 2\epsilon_f \leq -c_1 \alpha \Psi \left\| g_k \right\|^2 + 2\epsilon_A\]

(147)

which rearranges to

\[c_1 \Psi \left\| g_k \right\|^2 + \frac{\epsilon}{2} \left\| e(x_k) \right\|^2 \leq \frac{\epsilon}{2} \left\| \nabla \phi(x_k) \right\|^2 + \frac{2}{\alpha} (\epsilon_A - \epsilon_f).
\]

(148)

As \( \epsilon_A - \epsilon_f > 0 \), it is clear that the right side of (148) can be made arbitrarily large by sending \( \alpha \to 0 \). Hence, the relaxed Armijo condition (38) will be satisfied for sufficiently small \( \alpha \) and the backtracking line search will always find an \( \alpha_k \) small enough to satisfy (38).

By Lemma 2, outside of \( \mathcal{N}_1(\psi, \Psi, A\Psi \bar{e}_g) \), one has \( \delta_k < \frac{\omega}{(1 + \delta_A)} \). Applying the triangle and reverse triangle inequalities to (32) gives

\[
\left\| \nabla \phi(x_k) \right\|^2 - \left\| e(x_k) \right\|^2 \leq \left\| \nabla \phi(x_k) \right\|^2 + \left\| \nabla \phi(x_k) + e(x_k) \right\|^2 \leq \left\| \nabla \phi(x_k) \right\|^2 + \left\| e(x_k) \right\|^2
\]

(149)

which can be written using the gradient noise to signal ratio \( \delta_k \) as
\[
(1 - \delta_k) \| \nabla \phi(x_k) \|_2 \leq \| g_k \|_2 \leq (1 + \delta_k) \| \nabla \phi(x_k) \|_2. \tag{150}
\]

Combining the definition of \( \delta_k \) (see (58) in Lemma 2), (150), and \( \delta_k < \frac{\psi}{(1+\lambda)^p} \) with (144) gives

\[
f(x_k - aH_k g_k) \leq f(x_k) - \frac{\alpha \varepsilon}{2} (1 - \delta_k^2) \| \nabla \phi(x_k) \|_2^2 + 2\bar{\varepsilon}_f. \tag{151}
\]

\[
f(x_k) - \frac{\alpha \varepsilon}{2} (1 - \delta_k^2) \| \nabla \phi(x_k) \|_2^2 + 2\bar{\varepsilon}_f \leq f(x_k) - \frac{\alpha \varepsilon}{2} (1 + \delta_k^2) \| g_k \|_2^2 + 2\bar{\varepsilon}_f \tag{152}
\]

\[
f(x_k) - \frac{\alpha \varepsilon}{2} (1 - \delta_k^2) \| g_k \|_2^2 + 2\bar{\varepsilon}_f = f(x_k) - \frac{\alpha \varepsilon}{2} (1 - \delta_k^2) \| g_k \|_2^2 + 2\bar{\varepsilon}_f \tag{153}
\]

\[
f(x_k - aH_k g_k) \leq f(x_k) - \frac{\alpha \varepsilon}{2} (1 - \delta_k) \| g_k \|_2^2 + 2\bar{\varepsilon}_f \leq f(x_k) - \frac{\alpha \varepsilon}{2} (1 + \delta_k) \| g_k \|_2^2 + 2\bar{\varepsilon}_f. \tag{154}
\]

Now, as \( \bar{\varepsilon}_f < \varepsilon_A \), the bound (154) above implies the bound (146) for any \( \alpha \leq \frac{\varepsilon}{\psi^2 M} \) if \( c_1 \leq \frac{\varepsilon}{2\psi} \left( \frac{\psi^2}{1+\lambda^p} \right) \). Since \( \alpha_k \) is chosen using a backtracking line search with backtracking factor \( \tau < 1 \), it is true that \( \frac{\tau \varepsilon}{\psi^2 M} < \alpha_k \leq \frac{\varepsilon}{\psi^2 M} \). Thus, combining the bound (146) with Assumption 1 and (150) shows that

\[
\phi(x_k - a_k H_k g_k) \leq \phi(x_k) - c_1 \alpha_k \Psi \| g_k \|_2^2 + 2\varepsilon_A + 2\bar{\varepsilon}_f. \tag{155}
\]

\[
\phi(x_k) - c_1 \alpha_k \Psi \| g_k \|_2^2 + 2\varepsilon_A + 2\bar{\varepsilon}_f \leq \phi(x_k) - \frac{c_1 \varepsilon}{\psi M} (1 - \delta_k^2) \| \nabla \phi(x_k) \|_2^2 + 2\varepsilon_A + 2\bar{\varepsilon}_f \tag{156}
\]

\[
\phi(x_k - a_k H_k g_k) \leq \phi(x_k) - \frac{c_1 \varepsilon}{\psi M} \left( 1 - \frac{\psi}{(1+\lambda)^p} \right)^2 \| \nabla \phi(x_k) \|_2^2 + 2\varepsilon_A + 2\bar{\varepsilon}_f. \tag{157}
\]

The expression (157) measures the reduction in the value of \( \phi \) for iterates where \( x_k \notin \mathcal{N}(\psi, \Psi, A\Psi \bar{\varepsilon}_f) \). Proceeding, we take the following bound

\[
\phi(x_{k+1}) \leq \phi(x_k) - \frac{c_1 \varepsilon}{\psi M} \left( 1 - \frac{\psi}{(1+\lambda)^p} \right)^2 \| \nabla \phi(x_k) \|_2^2 + 2\varepsilon_A + 2\bar{\varepsilon}_f. \tag{158}
\]

and subtract \( \phi^* \) from both sides as well as apply the inequality (140) to get

\[
\phi(x_{k+1}) - \phi^* \leq \left( 1 - \frac{2mc_1 \varepsilon (1 - \frac{\psi}{(1+\lambda)^p})^2}{\psi M} \right) (\phi(x_k) - \phi^*) + 2\varepsilon_A + 2\bar{\varepsilon}_f. \tag{159}
\]
For ease of notation, define the following quantities
\[
\rho := \left(1 - \frac{2mc_1 \tau \varepsilon}{\Psi M} \left(1 - \frac{\Psi}{(1+A)\Psi}\right)^2\right), \quad \eta := 2\varepsilon_A + 2\bar{\varepsilon}_f. \tag{160}
\]
Thus, for all \( k \) where \( x_k \notin N_1(\psi, \Psi, A\Psi \bar{\varepsilon}_g) \), we have shown that the following bound holds
\[
\phi(x_{k+1}) - \phi^* \leq \rho(\phi(x_k) - \phi^*) + \eta. \tag{161}
\]
Subtracting \( \frac{\eta}{1-\rho} \) from both sides shows that
\[
\phi(x_{k+1}) - \phi^* - \frac{\eta}{1-\rho} \leq \rho(\phi(x_k) - \phi^*) + \eta - \frac{\eta}{1-\rho} = \rho(\phi(x_k) - \phi^*) - \frac{\rho\eta}{1-\rho} = \rho(\phi(x_k) - \phi^* - \frac{\eta}{1-\rho})
\]
and thus one has
\[
\phi(x_{k+1}) - \phi^* - \bar{\eta} \leq \rho(\phi(x_k) - \phi^* - \bar{\eta}) \tag{162}
\]
where \( \bar{\eta} := \frac{\eta}{1-\rho} \). Using the definitions in (160) shows that
\[
\bar{\eta} = \frac{\Psi M}{2mc_1 \tau \varepsilon} \left(1 - \frac{\Psi}{(1+A)\Psi}\right)^2 \left(2\varepsilon_A + 2\bar{\varepsilon}_f\right) \tag{163}
\]
\[
\bar{\eta} = \frac{\Psi M}{mc_1 \tau \varepsilon} \left(1 - \frac{\Psi}{(1+A)\Psi}\right)^2 (\varepsilon_A + \bar{\varepsilon}_f) \tag{164}
\]
which establishes (68) and (69). Similar to Appendix 7, we obtain the R-linear result (71) by recursively applying the bound in (68), stopping once an iterate enters \( N_1(\psi, \Psi, A\Psi \bar{\varepsilon}_g) \). This concludes the proof.

### Appendix 9: Extended numerical experiments

Table 5 below shows the performance of gradient descent for the same problem (ROSENBR) and noise combinations as in Table 1.

Tables 6, 7 and 8 compare the performance of SP-BFGS, BFGS, and gradient descent on the 32 CUTEst test problems with only gradient noise present (i.e. \( \bar{\varepsilon}_f = 0 \)). Gradient noise was generated using \( \bar{\varepsilon}_g = 10^{-4} \| \nabla \phi(x^0) \|_2 \), where the starting
Table 5  Performance of gradient descent on the Rosenbrock function (i.e. ROSENBR) corrupted by noise

| \( \epsilon_f \) | \( \epsilon_g \) | Mean (\( \Delta_{opt} \)) | Median (\( \Delta_{opt} \)) | Min (\( \Delta_{opt} \)) | Max (\( \Delta_{opt} \)) | \( s^2(\Delta_{opt}) \) | Mean(I) |
|---|---|---|---|---|---|---|---|
| \( 10^{-4} \) | \( 10^{-4} \) | \(-2.5E+00\) | \(-2.5E+00\) | \(-2.5E+00\) | \(-2.4E+00\) | \(2.3E-06\) | 204 |
| \( 10^{-2} \) | \( 10^{-2} \) | \(-2.3E+00\) | \(-2.3E+00\) | \(-2.9E+00\) | \(-2.3E+00\) | \(8.6E-02\) | 204 |
| \( 10^{0} \) | \( 10^{0} \) | \(-1.7E+00\) | \(-1.6E+00\) | \(-5.1E+00\) | \(+5.2E-01\) | \(1.8E+00\) | 107 |

Gradient descent with low function noise level

| \( 10^{-4} \) | \( 10^{-4} \) | \(-2.5E+00\) | \(-2.5E+00\) | \(-2.5E+00\) | \(-2.4E+00\) | \(1.6E-06\) | 205 |
| \( 10^{-2} \) | \( 10^{-2} \) | \(-2.5E+00\) | \(-2.5E+00\) | \(-2.9E+00\) | \(-2.3E+00\) | \(2.4E-02\) | 204 |
| \( 10^{0} \) | \( 10^{0} \) | \(-3.2E+00\) | \(-3.5E+00\) | \(-5.9E+00\) | \(-8.3E-01\) | \(1.7E+00\) | 195 |
| \( 10^{2} \) | \( 10^{2} \) | \(-2.8E+00\) | \(-2.8E+00\) | \(-5.3E+00\) | \(-9.1E-01\) | \(1.5E+00\) | 119 |

Gradient descent with medium function noise level

| \( 10^{-4} \) | \( 10^{-4} \) | \(-2.4E+00\) | \(-2.6E+00\) | \(-3.2E+00\) | \(-1.2E+00\) | \(3.1E-01\) | 201 |
| \( 10^{-2} \) | \( 10^{-2} \) | \(-2.5E+00\) | \(-2.6E+00\) | \(-2.8E+00\) | \(-1.2E+00\) | \(1.3E-01\) | 202 |
| \( 10^{0} \) | \( 10^{0} \) | \(-1.3E+00\) | \(-1.0E+00\) | \(-4.2E+00\) | \(-6.5E-01\) | \(7.1E-01\) | 194 |
| \( 10^{2} \) | \( 10^{2} \) | \(-2.3E+00\) | \(-2.6E+00\) | \(-3.6E+00\) | \(-4.0E-01\) | \(6.1E-01\) | 147 |

Gradient descent with high function noise level

| \( 10^{0} \) | \( 10^{0} \) | \(-5.2E-01\) | \(-4.9E-01\) | \(-6.8E-01\) | \(-4.1E-01\) | \(5.5E-03\) | 186 |
| \( 10^{0} \) | \( 10^{0} \) | \(-5.2E-01\) | \(-5.0E-01\) | \(-6.6E-01\) | \(-3.8E-01\) | \(5.8E-03\) | 186 |
| \( 10^{0} \) | \( 10^{0} \) | \(-5.4E-01\) | \(-5.4E-01\) | \(-6.7E-01\) | \(-4.1E-01\) | \(6.6E-03\) | 187 |
| \( 10^{2} \) | \( 10^{2} \) | \(-1.5E+00\) | \(-1.4E+00\) | \(-3.0E+00\) | \(-4.6E-01\) | \(2.9E-01\) | 195 |

\( \Delta_{opt} := \log_{10}(\phi_{best} - \phi^*) \) measures the optimality gap, where \( \phi_{best} \) denotes the smallest value of the true function \( \phi \) measured at any point during an algorithm run. The number of objective function evaluations is fixed at 2000, but the number of iterations \( I \) can vary. Statistics are calculated from a sample of 30 runs per algorithm.

Point \( x^0 \) varies by CUTEst problem, to ensure that noise does not initially dominate gradient evaluations. By examining the mean and median columns in Tables 6, 7 and 8, one sees that SP-BFGS outperforms both BFGS and gradient descent on \( \frac{18}{32} \approx 56\% \) of the CUTEst problems with only gradient noise present, and performs at least as well as the best performing alternative on \( \frac{28}{32} \approx 88\% \) of these problems. Equivalently, SP-BFGS was only outperformed by BFGS or gradient descent on \( \frac{4}{32} \approx 12\% \) of these problems.
Table 6 Performance of SP-BFGS on 32 selected CUTEst test problems with noise added to gradient evaluations only (i.e. $\bar{\epsilon}_f = 0$)

| Problem       | Dim. | Mean ($\Delta_{opt}$) | Median ($\Delta_{opt}$) | Min ($\Delta_{opt}$) | Max ($\Delta_{opt}$) | $s^2$($\Delta_{opt}$) |
|---------------|------|-----------------------|------------------------|---------------------|---------------------|----------------------|
| **SP-BFGS with gradient noise only** |      |                       |                        |                     |                     |                      |
| ARGTRIGLS     | 200  | -9.6E-02              | -9.6E-02               | -1.0E-01            | -8.5E-02            | 1.9E-05              |
| ARWHEAD       | 500  | -2.8E+00              | -2.8E+00               | -2.8E+00            | -2.7E+00            | 1.7E-03              |
| BEALE         | 2    | -1.4E+01              | -1.4E+01               | -1.6E+01            | -7.0E+00            | 4.1E+00              |
| BOX3          | 3    | -6.7E+00              | -6.5E+00               | -1.1E+01            | -6.3E+00            | 6.3E-01              |
| BOXPOWER      | 100  | -2.7E+00              | -2.7E+00               | -3.1E+00            | -2.3E+00            | 4.6E-02              |
| BROWNB5       | 2    | -4.5E+00              | -5.9E+00               | -8.0E+00            | +1.1E+00            | 8.4E+00              |
| BROYDNBDLS    | 50   | -5.4E+00              | -5.4E+00               | -5.9E+00            | -5.0E+00            | 3.4E-02              |
| CHAINWOO      | 100  | +1.6E+00              | +1.7E+00               | +7.6E-02            | +2.1E+00            | 1.5E-01              |
| CHNROSNB      | 50   | -3.2E+00              | -3.0E+00               | -4.9E+00            | -2.6E+00            | 4.5E-01              |
| COATING       | 134  | +3.4E-01              | +3.4E-01               | +1.8E-01            | +4.2E-01            | 3.1E-03              |
| COOLHANSLS    | 9    | -9.4E-01              | -9.4E-01               | -1.2E+00            | -4.8E-01            | 4.2E-02              |
| CUBE          | 2    | -2.7E+00              | -2.5E+00               | -5.8E+00            | -1.7E+00            | 7.5E-01              |
| CYCLOOCFLS    | 20   | -7.4E+00              | -7.2E+00               | -9.3E+00            | -5.9E+00            | 8.1E-01              |
| EXTROSNB      | 10   | -5.1E+00              | -5.2E+00               | -5.3E+00            | -4.7E+00            | 3.0E-02              |
| FMINSRF2      | 64   | -8.6E+00              | -8.7E+00               | -8.8E+00            | -8.1E+00            | 3.4E-02              |
| GENHUMPS      | 5    | -2.7E+00              | -2.6E+00               | -5.2E+00            | -1.0E+00            | 1.1E+00              |
| GENROSE       | 5    | -1.2E+01              | -1.2E+01               | -1.4E+01            | -8.9E+00            | 2.0E+00              |
| HEART6LS      | 6    | +1.0E+00               | +1.2E+00               | -1.8E+00            | +1.2E+00            | 5.0E-01              |
| HELIX         | 3    | -5.7E+00              | -5.9E+00               | -8.7E+00            | -3.4E+00            | 1.4E+00              |
| MANCINO       | 30   | -1.0E+00              | -1.0E+00               | -1.4E+00            | -7.0E-01            | 3.7E-02              |
| METHANB8LS    | 31   | -3.6E+00              | -3.6E+00               | -4.0E+00            | -3.3E+00            | 3.1E-02              |
| MODBEALE      | 200  | +1.2E+00              | +1.2E+00               | +3.8E-01            | +1.9E+00            | 1.8E-01              |
| NONDIA        | 10   | -3.5E-03              | -3.6E-03               | -4.3E-03            | -1.1E-03            | 6.6E-07              |
| POWELLSG      | 4    | -5.7E+00              | -5.3E+00               | -9.3E+00            | -4.0E+00            | 1.6E+00              |
| POWER         | 10   | -3.5E+00              | -3.5E+00               | -4.4E+00            | -2.8E+00            | 1.3E-01              |
| ROSENBR       | 2    | -1.1E+01              | -1.2E+01               | -1.4E+01            | -5.1E+00            | 4.4E+00              |
| ROSENBRTU     | 2    | -1.9E+01              | -1.9E+01               | -2.2E+01            | -1.7E+01            | 1.1E+00              |
| SBRYBND       | 500  | +3.9E+00              | +3.9E+00               | +3.9E+00            | +3.9E+00            | 2.0E-05              |
| SINEVAL       | 2    | -1.3E+01              | -1.3E+01               | -1.8E+01            | -1.1E+01            | 3.3E+00              |
| SNAIL         | 2    | -1.5E+01              | -1.6E+01               | -1.8E+01            | -1.2E+01            | 1.6E+00              |
| SROSENBR      | 1000 | -9.7E-01              | -9.7E-01               | -1.3E+00            | -4.8E-01            | 3.2E-02              |
| VIBRBEAM      | 8    | +1.6E+00              | +1.6E+00               | +1.2E+00            | +2.8E+00            | 9.1E-02              |

The number of objective function evaluations is fixed at 2000. $\Delta_{opt} := \log_{10}(\phi_{best} - \phi^*)$ measures the optimality gap, where $\phi_{best}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. The SP-BFGS penalty parameter was set as $\beta_k = \frac{10^8}{\bar{\epsilon}_g} \| s_k \|_2 + 10^{-10}$. For each problem, gradient noise was generated using $\bar{\epsilon}_g = 10^{-4} \| \nabla \phi(x^0) \|_2$, where the starting point $x^0$ varies by CUTEst problem.
Table 7  Performance of BFGS on 32 selected CUTEst test problems with noise added to gradient evaluations only (i.e. $\bar{\epsilon}_f = 0$)

| Problem         | Dim. | Mean ($\Delta_{\text{opt}}$) | Median ($\Delta_{\text{opt}}$) | Min ($\Delta_{\text{opt}}$) | Max ($\Delta_{\text{opt}}$) | $\hat{s}^2(\Delta_{\text{opt}})$ |
|-----------------|------|------------------------------|-------------------------------|-----------------------------|-----------------------------|---------------------------------|
| BFGS with gradient noise only |
| ARGTRIGLS       | 200  | −9.2E−02                     | −9.3E−02                      | −9.9E−02                    | −8.2E−02                    | 1.7E−05                         |
| ARWHEAD         | 50   | −2.5E+00                     | −2.5E+00                      | −2.6E+00                    | −2.5E+00                    | 7.6E−04                         |
| BEALE           | 2    | −8.3E+00                     | −8.5E+00                      | −1.2E+01                    | −5.8E+00                    | 3.9E+00                         |
| BOX3            | 3    | −6.4E+00                     | −6.4E+00                      | −6.6E+00                    | −6.3E+00                    | 2.3E−03                         |
| BOXPOWER        | 100  | −2.8E+00                     | −2.8E+00                      | −3.3E+00                    | −2.4E+00                    | 6.1E−02                         |
| BROWNBS         | 2    | +6.8E−02                     | +1.0E+00                      | −8.2E+00                    | +3.6E+00                    | 1.0E+01                         |
| BROYDNBDLS      | 50   | −5.1E+00                     | −5.1E+00                      | −5.3E+00                    | −4.9E+00                    | 1.5E−02                         |
| CHAINWOO        | 100  | +1.7E+00                     | +1.8E+00                      | +1.1E+00                    | +2.2E+00                    | 5.8E−02                         |
| CHNROSNB        | 50   | −2.9E+00                     | −2.7E+00                      | −4.5E+00                    | −2.1E+00                    | 3.8E−01                         |
| COATING         | 134  | +3.6E−01                     | +3.7E−01                      | +2.1E−01                    | +4.2E−01                    | 2.6E−03                         |
| COOLHANSLS      | 9    | −5.6E−01                     | −6.4E−01                      | −1.3E+00                    | +1.9E−01                    | 1.9E−01                         |
| CUBE            | 2    | −1.1E+00                     | −1.1E+00                      | −1.8E+00                    | −9.6E−01                    | 5.6E−02                         |
| CYCLOOCFLS      | 20   | −6.5E+00                     | −6.5E+00                      | −8.3E+00                    | −5.1E+00                    | 5.4E−01                         |
| EXTROSNB        | 10   | −5.1E+00                     | −5.1E+00                      | −5.3E+00                    | −4.9E+00                    | 8.1E−03                         |
| FMINSRF2        | 64   | −8.2E+00                     | −8.2E+00                      | −8.7E+00                    | −7.3E+00                    | 1.4E−01                         |
| GENHUMPS        | 5    | −1.5E+00                     | −1.2E+00                      | −4.0E+00                    | −2.8E−01                    | 8.4E−01                         |
| GENROSE         | 5    | −6.8E+00                     | −6.7E+00                      | −8.7E+00                    | −5.9E+00                    | 4.8E−01                         |
| HEART6LS        | 6    | +1.2E+00                     | +1.2E+00                      | +1.2E+00                    | +1.2E+00                    | 1.9E−04                         |
| HELIX           | 3    | −4.8E+00                     | −4.6E+00                      | −8.1E+00                    | −2.6E+00                    | 2.1E+00                         |
| MANCINO         | 30   | −8.3E−01                     | −8.8E−01                      | −1.2E+00                    | −3.3E−01                    | 5.3E−02                         |
| METHANB8LS      | 31   | −3.5E+00                     | −3.4E+00                      | −3.9E+00                    | −3.3E+00                    | 2.8E−02                         |
| MODBEALE        | 200  | +1.0E+00                     | +1.1E+00                      | −6.2E−01                    | +2.1E+00                    | 3.6E−01                         |
| NONDIA          | 10   | +1.2E−03                     | +1.3E−03                      | −4.4E−03                    | +1.3E−02                    | 2.2E−05                         |
| POWELLSG        | 4    | −5.3E+00                     | −5.0E+00                      | −8.0E+00                    | −3.6E+00                    | 1.6E+00                         |
| POWER           | 10   | −3.4E+00                     | −3.4E+00                      | −4.3E+00                    | −2.8E+00                    | 1.4E−01                         |
| ROSENBR         | 2    | −6.1E+00                     | −5.9E+00                      | −1.0E+01                    | −3.7E+00                    | 2.9E+00                         |
| ROSENBRTL       | 2    | −1.5E+01                     | −1.5E+01                      | −1.8E+01                    | −1.4E+01                    | 1.6E+00                         |
| SBRYBND         | 500  | +3.9E+00                     | +3.9E+00                      | +3.9E+00                    | +3.9E+00                    | 2.0E−05                         |
| SINEVAL         | 2    | −1.2E+01                     | −1.3E+01                      | −1.7E+01                    | −8.5E+00                    | 4.0E+00                         |
| SNAIL           | 2    | −1.1E+01                     | −1.1E+01                      | −1.6E+01                    | −8.2E+00                    | 3.5E+00                         |
| SROSENBR        | 1000 | −9.1E−01                     | −8.8E−01                      | −1.3E+00                    | −5.1E−01                    | 3.1E−02                         |
| VIBRBEAM        | 8    | +1.7E+00                     | +1.6E+00                      | +1.4E+00                    | +2.6E+00                    | 1.0E−01                         |

The number of objective function evaluations is fixed at 2000. $\Delta_{\text{opt}} := \log_{10}(\phi_{\text{best}} - \phi^*)$ measures the optimality gap, where $\phi_{\text{best}}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. For each problem, gradient noise was generated using $\bar{\epsilon}_g = 10^{-4} || \nabla \phi(x^0) ||^2_2$, where the starting point $x^0$ varies by CUTEst problem.
Table 8 Performance of gradient descent on 32 selected CUTEst test problems with noise added to gradient evaluations only (i.e. $\bar{\varepsilon}_f = 0$)

| Problem       | Dim. | Mean ($\Delta_{opt}$) | Median ($\Delta_{opt}$) | Min ($\Delta_{opt}$) | Max ($\Delta_{opt}$) | $\sigma^2(\Delta_{opt})$ |
|---------------|------|------------------------|------------------------|---------------------|---------------------|-------------------|
| Gradient descent with gradient noise only |
| ARGTRIGLS     | 200  | -2.4E+00               | -2.4E+00               | -2.4E+00            | -2.3E+00            | 4.6E-05          |
| ARWHEAD       | 500  | -2.8E+00               | -2.8E+00               | -2.9E+00            | -2.8E+00            | 9.0E-05          |
| BEALE         | 2    | -9.1E+00               | -9.0E+00               | -1.4E+01            | -6.5E+00            | 2.8E+00          |
| BOX3          | 3    | -6.3E+00               | -6.3E+00               | -6.4E+00            | -6.2E+00            | 1.0E-03          |
| BOXPOWER      | 100  | -1.0E+00               | -1.0E+00               | -1.0E+00            | -9.9E-01            | 1.2E-06          |
| BROWNBS       | 2    | +1.2E+01               | +1.2E+01               | +3.0E+00            | +1.2E+01            | 2.7E+00          |
| BROYDNBDLS    | 50   | -5.8E+00               | -5.8E+00               | -6.1E+00            | -5.6E+00            | 2.1E-02          |
| CHAINWOO      | 100  | +2.6E+00               | +2.6E+00               | +2.6E+00            | +2.6E+00            | 7.5E-09          |
| CHNROSNB      | 50   | +1.6E+00               | +1.6E+00               | +1.5E+00            | +1.6E+00            | 1.9E-06          |
| COATING       | 134  | +2.5E+00               | +2.5E+00               | +2.5E+00            | +2.6E+00            | 5.1E-04          |
| COOLHANSLS    | 9    | -3.1E-01               | -2.1E-01               | -1.3E+00            | +9.6E-02            | 1.1E-01          |
| CUBE          | 2    | -2.5E+00               | -2.2E+00               | -6.7E+00            | -4.1E-01            | 2.0E+00          |
| CYCLOOCFLS    | 20   | -3.2E+00               | -4.2E+00               | -4.6E+00            | -2.2E-01            | 3.3E+00          |
| EXTRSNB       | 10   | -1.9E+00               | -1.9E+00               | -1.9E+00            | -1.9E+00            | 4.0E-04          |
| FMINSRF2      | 64   | -1.4E+00               | -1.4E+00               | -1.4E+00            | -1.4E+00            | 3.1E-09          |
| GENHUMPS      | 5    | -9.0E-01               | -9.1E-01               | -1.4E+00            | -3.2E-01            | 7.6E-02          |
| GENROSE       | 5    | -2.1E-01               | -2.1E-01               | -2.2E-01            | -2.0E-01            | 4.5E-05          |
| HEART6LS      | 6    | +1.5E+00               | +1.5E+00               | +1.5E+00            | +1.5E+00            | 3.3E-07          |
| HELIX         | 3    | -6.7E-02               | -6.8E-02               | -8.3E-02            | -3.4E-02            | 9.8E-05          |
| MANKINO       | 30   | -1.7E+00               | -1.7E+00               | -2.1E+00            | -1.5E+00            | 1.7E-02          |
| METHANB8LS    | 31   | -1.7E+00               | -1.7E+00               | -1.7E+00            | -1.7E+00            | 4.4E-09          |
| MODBEALE      | 200  | +2.3E+00               | +2.3E+00               | +2.3E+00            | +2.3E+00            | 3.0E-08          |
| NONDIA        | 10   | +3.9E-01               | +3.9E-01               | +3.8E-01            | +4.0E-01            | 2.1E-05          |
| POWELLSG      | 4    | -2.5E+00               | -2.4E+00               | -2.8E+00            | -2.4E+00            | 5.2E-03          |
| POWER         | 10   | -5.1E+00               | -4.9E+00               | -7.0E+00            | -3.9E+00            | 4.4E-01          |
| ROSENBR       | 2    | -3.4E+00               | -2.8E+00               | -7.7E+00            | -2.3E+00            | 2.0E+00          |
| ROSENBRTU     | 2    | -4.5E-02               | -2.4E-02               | -7.2E-02            | -2.1E-02            | 5.8E-04          |
| SBRYBND       | 500  | +3.9E+00               | +3.9E+00               | +3.9E+00            | +3.9E+00            | 9.5E-06          |
| SINEVAL       | 2    | +2.3E-01               | +2.3E-01               | +2.3E-01            | +2.3E-01            | 6.3E-09          |
| SNAIL         | 2    | -1.6E+01               | -1.6E+01               | -1.7E+01            | -1.5E+01            | 3.0E-01          |
| SROSENBR      | 1000 | -4.1E-01               | -4.1E-01               | -4.1E-01            | -4.1E-01            | 1.6E-06          |
| VIBRBEAM      | 8    | +3.0E+00               | +3.0E+00               | +2.9E+00            | +3.2E+00            | 6.6E-03          |

The number of objective function evaluations is fixed at 2000. $\Delta_{opt} := \log_{10}(\phi_{\text{best}} - \phi^*)$ measures the optimality gap, where $\phi_{\text{best}}$ denotes the smallest value of the true function $\phi$ measured at any point during an algorithm run. Statistics are calculated from a sample of 30 runs per algorithm, and the Dim. column gives the problem dimension. For each problem, gradient noise was generated using $\bar{\varepsilon}_f = 10^{-4}\|\nabla \phi(x^0)\|_2$, where the starting point $x^0$ varies by CUTEst problem.
Acknowledgements EH and BI’s work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of British Columbia (UBC).

Data availability The CUTEst test problems used in the numerical experiments are available at https://www.cuter.rl.ac.uk/Problems/mastsif.shtml.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aydin, L., Aydin, O., Artem, H.S., Mert, A.: Design of dimensionally stable composites using efficient global optimization method. Proc. Inst. Mech. Eng. Part L: J. Mater. Design Appl. 233(2), 156–168 (2019). https://doi.org/10.1177/1464420716664921
2. Berahas, A.S., Byrd, R.H., Nocedal, J.: Derivative-free optimization of noisy functions via quasi-newton methods. SIAM J. Optim. 29, 965–993 (2019). https://doi.org/10.1137/18M1177718
3. Bensen, M., Anthoff, D., Arslan, A., Byrne, S., Lin, D., Papamarkou, T., Pearson, J.: Distributions.jl: Definition and modeling of probability distributions in the juliastats ecosystem. arXiv e-prints arXiv:1907.08611 (2019)
4. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. SIAM Rev. 59(1), 65–98 (2017). https://doi.org/10.1137/141000671
5. Bons, N.P., He, X., Mader, C.A., Martins, J.R.R.A.: Multimodality in aerodynamic wing design optimization. AIAA J. 57(3), 1004–1018 (2019). https://doi.org/10.2514/1.J057294
6. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms 1. General considerations.IMA J. Appl. Math. 6(1), 76–90 (1970). https://doi.org/10.1093/imamat/6.1.76
7. Byrd, R.H., Hansen, S.L., Nocedal, J., Singer, Y.: A stochastic quasi-newton method for large-scale optimization. SIAM J. Optim. 26(2), 1008–1031 (2016). https://doi.org/10.1137/140954362
8. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. 16(5), 1190–1208 (1995). https://doi.org/10.1137/0916069
9. Byrd, R.H., Nocedal, J.: A tool for the analysis of quasi-newton methods with application to unconstrained minimization. SIAM J. Numer. Anal. 26(3), 727–739 (1989). https://doi.org/10.1023/0003-6811(2019)0.102076
10. Byrd, R.H., Nocedal, J., Yuan, Y.X.: Global convergence of a class of quasi-newton methods on convex problems. SIAM J. Numer. Anal. 24(5), 1171–1190 (1987). https://doi.org/10.2307/2157646
11. Chang, D., Sun, S., Zhang, C.: An accelerated linearly convergent stochastic L-BFGS algorithm. IEEE Trans. Neural Netw. Learn. Syst. 30(11), 3338–3346 (2019). https://doi.org/10.1109/TNNLS.2019.2891088
12. Fasano, G., Pintér, J.D.: Modeling and Optimization in Space Engineering: State of the Art and New Challenges. Springer (2019). https://doi.org/10.1007/978-1-4614-4469-5
13. Fletcher, R.: A new approach to variable metric algorithms. Comput. J. 13(3), 317–322 (1970). https://doi.org/10.1093/comjnl/13.3.317
14. Gal, R., Haber, E., Irwin, B., Saleh, B., Ziv, A.: How to catch a lion in the desert: on the solution of the coverage directed generation (CDG) problem. Optim. Eng. 22, 217–245 (2021). https://doi.org/10.1007/s11081-020-09507-w
15. Goldfarb, D.: A family of variable-metric methods derived by variational means. Math. Comput. 24(109), 23–26 (1970). https://doi.org/10.1090/s1088-6812-70-004873
16. Gould, N.I.M., Orban, D., contributors: The Constrained and Unconstrained Testing Environment with safe threads (CUTEst) for optimization software. https://github.com/ralna/CUTEst (2019)
17. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEr a Constrained and Unconstrained Testing Environment, revisited. https://www.cuter.rl.ac.uk (2001)
18. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEr and SifDec: a constrained and unconstrained testing environment, revisited. ACM Trans. Math. Softw. 29(4), 373–394 (2003). https://doi.org/10.1145/962437.962439
Secant penalized BFGS: a noise robust quasi-Newton method…

19. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput. Optim. Appl. 60(3), 545–557 (2015). https://doi.org/10.1007/s10589-014-9687-3

20. Gower, R., Goldfarb, D., Richtarik, P.: Stochastic block BFGS: squeezing more curvature out of data. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1869–1878. PMLR, New York, New York, USA (2016). http://proceedings.mlr.press/v48/gower16.html

21. Graf, P.A., Billups, S.: MDTri: robust and efficient global mixed integer search of spaces of multiple ternary alloys. Comput. Optim. Appl. 68(3), 671–687 (2017). https://doi.org/10.1007/s10589-017-9922-9

22. Güler, O., Gürtuna, F., Shevchenko, O.: Duality in quasi-newton methods and new variational characterizations of the DFP and BFGS updates. Optim. Methods Softw. 24(1), 45–62 (2009). https://doi.org/10.1080/10556780802367205

23. Hager, W.W.: Updating the inverse of a matrix. SIAM Review 31(2), 221–239 (1989). https://doi.org/10.2307/2030425

24. Horn, R.A., Johnson, C.R.: Matrix Analysis, 2nd edn. Cambridge University Press, New York (2013). https://doi.org/10.1017/CBO9781139020411

25. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, pp. 315–323 (2013). https://doi.org/10.5555/2999611.2999647

26. Johnson, S.G.: Quasi-newton optimization: Origin of the BFGS update (2019). https://ocw.mit.edu/courses/mathematics/18-335j-introduction-to-numerical-methods-spring-2019/week-1/MIT18_335JS19_lec30.pdf

27. Keane, A.J., Nair, P.B.: Computational Approaches for Aerospace Design: The Pursuit of Excellence. Wiley (2005). https://doi.org/10.1002/0470855487

28. Kelley, C.: Implicit Filtering. SIAM, Philadelphia (2011). https://doi.org/10.1137/1.9781611971903

29. Koziel, S., Ogurtsov, S.: Antenna Design by Simulation-Driven Optimization. Springer (2014). https://doi.org/10.1007/978-3-319-04367-8

30. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-newton methods. Math. Program. 141, 135–163 (2013). https://doi.org/10.1007/s10107-012-0514-2

31. Lin, D., White, J.M., Byrne, S., Bates, D., Noack, A., Pearson, J., Arslan, A., Squire, K., Anthoff, D., Papamarkou, T., Besançon, M., Drugowitsch, J., Schauer, M., other contributors: JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions. https://github.com/JuliaStats/Distributions.jl (2019). https://doi.org/10.5281/zenodo.2647458

32. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. 50(1), 503–528 (1998). https://doi.org/10.1007/BF01589116

33. Mokhtari, A., Ribeiro, A.: Global convergence of online limited memory BFGS. J. Mach. Learn. Res. 16(1), 3151–3181 (2015). https://doi.org/10.5555/2789272.2912100

34. Moritz, P., Nishihara, R., Jordan, M.: A linearly-convergent stochastic L-BFGS algorithm. In: Gretton, A., Robert, C.C. (eds.) Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 51, pp. 249–258. PMLR, Cadiz, Spain (2016). http://proceedings.mlr.press/v51/moritz16.html

35. Muñoz-Rojas, P.A.: Computational Modeling, Optimization and Manufacturing Simulation of Advanced Engineering Materials. Springer (2016). https://doi.org/10.1007/978-3-319-04265-7

36. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40406-5

37. Orban, D., Siqueira, A.S., contributors: CUTEst.jl: Julia’s CUTest interface. https://github.com/JuliaSmoothOptimizers/CUTEst.jl (2020). https://doi.org/10.5281/zenodo.1188851

38. Orban, D., Siqueira, A.S., contributors: NLPModels.jl: Data structures for optimization models. https://github.com/JuliaSmoothOptimizers/NLPModels.jl (2020). https://doi.org/10.5281/zenodo.2558627

39. Powell, M.J.D.: Algorithms for nonlinear constraints that use lagrangian functions. Math. Program. 14(1), 224–248 (1978). https://doi.org/10.1007/BF01588967

40. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. Comput. J. 3(3), 175–184 (1960). https://doi.org/10.1093/comjnl/3.3.175

41. Schraudolph, N.N., Yu, J., Günter, S.: A stochastic quasi-newton method for online convex optimization. In: Meila, M., Shen, X. (eds.) Proceedings of the Eleventh International Conference on
Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 2, pp. 436–443. PMLR, San Juan, Puerto Rico (2007). http://proceedings.mlr.press/v2/schraudolph07a.html

42. Shanno, D.F.: Conditioning of quasi-newton methods for function minimization. Math. Comput. 24(111), 647–656 (1970). https://doi.org/10.2307/2004840

43. Shi, H.J.M., Xie, Y., Byrd, R., Nocedal, J.: A noise-tolerant quasi-newton algorithm for unconstrained optimization. SIAM J. Optim. 32(1), 29–55 (2022). https://doi.org/10.1137/20M1373190

44. Wang, X., Ma, S., Goldfarb, D., Liu, W.: Stochastic quasi-newton methods for nonconvex stochastic optimization. SIAM J. Optim. 27(2), 927–956 (2017). https://doi.org/10.1137/15M1053141

45. Xie, Y., Byrd, R.H., Nocedal, J.: Analysis of the BFGS method with errors. SIAM J. Optim. 30(1), 182–209 (2020). https://doi.org/10.1137/19M1240794

46. Zhao, R., Haskell, W.B., Tan, V.Y.F.: Stochastic L-BFGS: improved convergence rates and practical acceleration strategies. IEEE Trans. Signal Process. 66, 1155–1169 (2018). https://doi.org/10.1109/TSP.2017.2784360

47. Zhu, J.: Optimization of Power System Operation. Wiley (2008). https://doi.org/10.1002/9780470466971

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.