Noise-Tolerant Optimization Methods for the Solution of a Robust Design Problem*

Yuchen Lou†, Shigeng Sun‡, and Jorge Nocedal†

Abstract. The development of nonlinear optimization algorithms capable of performing reliably in the presence of noise has garnered considerable attention lately. This paper advocates for strategies to create noise-tolerant nonlinear optimization algorithms by adapting classical deterministic methods. These adaptations follow certain design guidelines described here, which make use of estimates of the noise level in the problem. The application of our methodology is illustrated by the development of a line search gradient projection method, which is tested on an engineering design problem. It is shown that a new self-calibrated line search and noise-aware finite-difference techniques are effective even in the high noise regime. Numerical experiments investigate the resiliency of key algorithmic components. A convergence analysis of the line search gradient projection method establishes convergence to a neighborhood of the solution.

Key words. Nonlinear optimization, gradient projection method, stochastic optimization, robust design.

MSC codes. 90C30, 90C15, 93B51, 65K05.

1. Introduction.

Over the past 50 years, significant progress has been made in the development of robust and efficient methods for deterministic nonlinear optimization. These methods have been adopted in a wide range of applications, and in the case of constrained optimization, can be quite complex. Recently, there has been a growing interest in tackling nonlinear problems where the function and/or gradient evaluations are subject to noise or errors [1, 4, 10, 11, 13, 19, 26, 29, 42]. This raises the question of whether existing optimization methods require substantial redesign to ensure robustness in the presence of noise, or if certain modifications are sufficient to tackle such challenges.

This paper argues that one can develop effective methods for a broad range of noisy optimization problems by retaining the fundamental properties of deterministic methods while incorporating certain modifications based on the design guidelines outlined herein. These guidelines stem from the observation that, in the presence of noise, only few operations can lead to numerical difficulties in optimization methods. These operations include:

1. Comparisons of noisy function values, as required e.g., in line search and trust region techniques.
2. Computation of differences of noisy function values, as required in finite-difference approximations to a gradient.
3. Computation of differences of noisy gradients, a basic ingredient in quasi-Newton updating.

Robust methods can be designed by ensuring that these operations are conducted reliably,

---

*Submitted to the editors Jan 17th, 2024.

Funding: This work was supported by National Science Foundation grant DMS-2011494, AFOSR grant FA95502110084, and ONR grant N00014-21-1-2675.

†Department of Industrial Engineering and Management Sciences, Northwestern University, USA.

‡Department of Engineering Sciences and Applied Mathematics, Northwestern University, USA.

preventing the algorithm from making harmful decisions. In this paper, we explore stabilization procedures that utilize an upper bound or a standard deviation of the noise (referred to as the *noise level*), and illustrate their performance in solving a design optimization problem. Examples of strategies proposed in the literature to safeguard the three fragile operations mentioned above are as follows. *Soft comparisons:* when assessing whether a step is acceptable by comparing noisy function values, the classical sufficient decrease condition can be relaxed in proportion to the noise level [2, 3, 39]; *Robust difference intervals:* in computing a finite difference gradient approximation, the distance between evaluation points for noisy functions should be proportional the square root of the noise level divided by the norm of the Hessian [26, 38]; *Controlled gradient differences:* quasi-Newton methods can achieve robustness by ensuring that points used for computing gradient differences (normally consecutive iterates) are adequately spaced in relation to the noise level in the problem [37, 42].

We do not argue that *the only* way to design nonlinear optimization methods for noisy problems is to adapt existing deterministic methods. We will see that in scenarios with highly noisy gradients, deviating from traditional approaches can be beneficial. Specifically, utilizing techniques like diminishing steplengths [6, 28, 33] can help counteract the adverse impacts of errors or noise, offering a viable alternative to line searches or trust region techniques. Nevertheless, the sophistication of some of the established methods and software for deterministic optimization makes it alluring to build upon their foundations as much as possible because of the important algorithmic ideas they embody. For example, in cases where a good estimate of the optimal active set is available, it is sensible to employ an active set method like sequential quadratic programming, as it can effectively utilize this estimate [34]. Similarly, primal-dual interior point methods have demonstrated remarkable efficacy in handling large-scale problems with network structure [20]. Maintaining these capabilities even amidst noise is highly desirable.

In this paper, we study the performance of an algorithm that follows the design principles mentioned above and apply it to a design optimization problem in which the noise level can be adjusted. In this problem, the goal is to optimize the shape of an acoustic horn to achieve optimal efficiency, assuming that there is uncertainty in some of the physical properties of the system [29]. This leads to a nonconvex bound constrained optimization problem, for which we design a noise-tolerant gradient projection method with a new *self-calibrated line search* that incorporates noise suppression within the classical framework. Our case study provides ample flexibility for assessing the efficacy of various optimization methods as noise increases from mild to extremely high, a regime where the stochastic gradient descent (SGD) method [33] has shown to be particularly effective.

1.1. Contributions of the Paper.

The recent literature on noisy nonlinear optimization typically reports numerical tests using either synthetic noise or simple machine learning models, leaving the question of their effectiveness in realistic applications open. In this paper, we focus on the sources of noise and errors that arise in certain practical problems, identify three critical operations prone to failure, and discuss the importance of the noise level in designing noise-tolerant algorithms. Based on a case study in optimal design, we conduct systematic tests to verify the robustness of two key components of our gradient projection method, namely the line search and the finite difference gradient approximation, as the noise level in the problem increases.

Building upon these findings, we introduce a new *self-calibrated line search* technique, effective even in environments with high levels of noise. This technique narrows the gap between traditional algorithms and the fixed step length SGD method. Additionally, we provide a convergence analysis for the line search gradient projection algorithm used in our case study, under the assumption that the noise in the function is bounded—a realistic assumption in this context.

1.2. Organization of the Paper. This paper is structured into seven sections. In the following section, we explore the concept of *noise level* and its estimation. Section §3 introduces the optimal design problem central to our study. In Section §4, we detail a gradient projection method rooted in robust design principles. Section §5 presents the results of our numerical tests, while Section §6 offers a global convergence analysis of the gradient projection method with a line search. The paper concludes with final remarks in Section §7.

2. Noise and Errors.

Let f be a smooth function and $\tilde{f}$ its noisy or inexact counterpart. Polyak [32] proposed two broad categories of noise and errors:

$$\tilde{f}(x) = f(x) + \Delta(x) \quad \text{stochastic noise,} \tag{2.1}$$

$$\tilde{f}(x) = f(x) + \delta(x) \quad \text{deterministic error.} \tag{2.2}$$

The first case arises e.g. from Monte Carlo simulation, and thus $\Delta(x) \sim D_x$ is a random variable following a distribution D_x that may be parameterized by x. The second case concerns computational error, broadly speaking, where repeated evaluations of $\tilde{f}(x)$ for a given x give the same result.

Following Moré and Wild [25, 26], we use the term *noise level* of a function. For the case of stochastic noise, we define the noise level of $\tilde{f}$ at a point x as the standard deviation of $\tilde{f}(x)$, which we denote $\sigma_f(x)$. In practice, we compute an estimate $\epsilon_f(x)$:

$$\epsilon_f(x) \approx \sigma_f(x) := \sqrt{\mathbb{V}(\tilde{f}(x))}. \tag{2.3}$$

There are situations where deterministic error (2.2) can be described in a useful manner using a stochastic model, so that $\delta(x)$ can be viewed as a realization of a random variable. In this case, we say that the function exhibits *computational noise*, and we will denote the resultant random variable as $\Delta(x)$, as in the case of stochastic noise. Following Moré and Wild [25, 26], we define the noise level $\sigma_f(x)$ as the standard deviation of $\Delta(x)$, with $\epsilon_f(x)$ serving as an approximate measure. For example, roundoff error is deterministic but can be modeled (albeit imperfectly) using a random variable drawn from a uniform distribution over the interval $[-|f(x)|\epsilon_M, |f(x)|\epsilon_M]$, where ϵ_M is unit roundoff. More examples of computational noise can be found in [25] and in §3.4 of this paper.

In summary, stochastic and computational noise can be analyzed using a uniform approach by studying the properties of $\Delta(x)$.

In the more general case of deterministic error, we can employ an estimate of the maximum error:

$$\epsilon_b \approx \sup |\delta(x)|, \quad x \in \mathcal{R}, \tag{2.4}$$

where $\mathcal{R}$ is the region of interest.

2.1. Noise Level Estimation. Knowledge of the noise level in the function is a key component in the algorithms described in this paper. As a result, we now discuss some practical procedures for estimating the noise level.

Local Pointwise Estimate $\epsilon_f(x)$. Given m i.i.d. samples $\{\tilde{f}_1(x), \tilde{f}_2(x), \ldots, \tilde{f}_m(x)\}$, we can define the pointwise noise level, in the case of stochastic noise, as

$$\epsilon_f(x) := \sqrt{\frac{1}{m-1}\sum_{j=1}^{m}\left(\tilde{f}_j(x) - \overline{\tilde{f}(x)}\right)^2}, \quad \text{where} \quad \overline{\tilde{f}(x)} := \frac{1}{m}\sum_{j=1}\tilde{f}_j(x). \tag{2.5}$$

From classic statistics, we know that $\epsilon_f(x)$ is an unbiased and consistent estimator of $\sigma_f(x) = [\mathbb{V}(\tilde{f}(x))]^{1/2}$.

We observe that formula (2.5) is not suitable in the context of computational noise. Since this type of noise is deterministic, the formula would erroneously suggest a noise level of zero. One can, however, use the `ECNoise` algorithm [25], which was specifically designed for computational noise. It samples points along a randomly chosen line and employs Hamming differences [21] to yield an estimate $\epsilon_f(x)$.

Global Estimate ϵ_f. Estimating $\epsilon_f(x_k)$ at every iteration is expensive and often unnecessary in practice. Whenever possible, it is desirable to employ a universal estimate ϵ_f for all x in the region of interest. A global measure of noise over the region of interest $\mathcal{R}$ can be defined as

$$\overline{\sigma}_f = \frac{1}{|\mathcal{R}|}\int_{\mathcal{R}} \sigma_f(x)dx, \tag{2.6}$$

and can be estimated as

$$\epsilon_f := \frac{1}{M}\sum_{i=1}^{M}\epsilon_f(x_i) \approx \overline{\sigma}_f \tag{2.7}$$

where $\{x_1, \cdots, x_M\}$ are randomly sampled from $\mathcal{R}$, and $\epsilon_f(x_i)$ is either given by (2.5) or is the output of `ECNoise`.

In some cases, e.g. Figure 3 in the next section, $\sigma_f(x)$ remains relatively constant across $\mathcal{R}$, allowing us to use a few (ideally only one representative) sample point x_i to define ϵ_f.

There are other more powerful estimators in the statistics literature but they are typically more expensive. Considering the iterative aspect of optimization algorithms, the simpler constant estimators ϵ_f defined above are often adequate for practical purposes, as illustrated in §5.

3. Case Study: An Acoustic Design Problem. To guide our discussion on the design of robust optimization methods and illustrate the concept of noise level, we begin by presenting a case study involving optimal design under uncertainty. In this problem, the uncertainty of some system parameters and the use of sampling techniques lead to noise in the objective function. While the uncertainty in the parameters is well-defined, predicting its propagation into the objective function becomes challenging owing to the nonlinear nature of the simulation. Nonetheless, we will see that estimating the noise level in the function is feasible, enabling us to effectively utilize a range of approaches to solve the optimization problem.

3.1. Statement of the Problem. We consider the 2-D acoustic design problem under uncertainty studied by Ng and Willcox [29]. An incoming wave enters a horn through its inlet and exits the outlet into the exterior domain with an absorbing boundary; see Figure 1. The goal is to find the shape of the horn so as to optimize its efficiency.

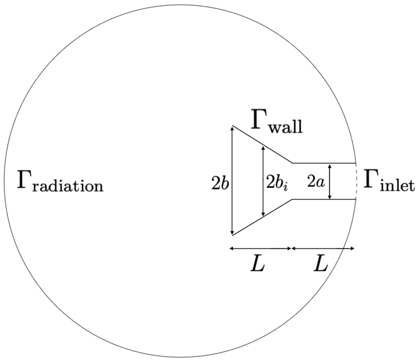

Figure 1: Schematic plot for the design of horn

The propagation of the acoustic wave is modeled by the non-dimensional complex Helmholtz equation

$$\nabla^2 u + \hat{k}^2 u = 0, \tag{3.1}$$

where u represents velocity and $\hat{k}$ is the wave number. The design variables $b = (b_1, b_2, \cdots, b_6)$ in $\mathbb{R}^6$ define the flare half-widths. We impose bounds on the design variables, $b_L \leq b \leq b_U$, and assume that the dimensions, a, L, depicted in Figure 1 are given. The PDE is solved using a finite element method.

The model contains uncertainties. The impedances z_l and z_u of the lower and upper horn walls are not known, but are assumed to follow a Gaussian distribution, $N(50, 3)$. Similarly, the wave number $\hat{k}$ is assumed to follow a uniform distribution Unif$(1.3, 1.5)$. We characterize uncertainty by the random variable ω, so that $\hat{k}(\omega) \sim \text{Unif}(1.3, 1.5)$, $z_l(\omega) \sim N(50, 3)$, and $z_u(\omega) \sim N(50, 3)$.

For a particular realization ξ_i of the random variable ω, the efficiency s of the horn is characterized by the flux at the inlet, as follows:

$$s(b, \xi_i) = \left| \int_{\Gamma_{\text{inlet}}} u(b, \xi_i, t) dt - 1 \right|. \tag{3.2}$$

Ng and Wilcox employ various statistics of $s(b, \omega)$ to estimate overall efficiency and to achieve a robust design. We focus here on the following formulation

$$\min_{b_L \leq b \leq b_U} f(b) = \mathbb{E}[s(b, \omega)] + 3\sqrt{\mathbb{V}[s(b, \omega)]}. \tag{3.3}$$

Although one may argue in favor of other robust formulations, the precise choice of the objective is not important in the discussion that follows. Note that problem (3.3) is a bound constrained stochastic optimization problem.

3.2. Approximating the Objective Function. Closed form representations of the expectation and variance terms in (3.3) are unknown and must be estimated by sampling. At every iteration k of the optimization algorithm, we compute the stochastic approximation:

$$\tilde{f}(b_k) = \bar{s}_k(b_k, \Xi_k) + 3\sqrt{S_k(b_k, \Xi_k)^2}, \tag{3.4}$$

where $\Xi_k = \{\xi_1, \xi_2, \cdots, \xi_N\}$ is a batch of i.i.d. samples of the random variable ω. Here, $\bar{s}_k(b_k, \Xi_k)$ is the sample mean of $s(b_k, \xi_i)$ with respect to the batch Ξ_k, i.e.,

$$\bar{s}_k(b_k, \Xi_k) = \frac{1}{N} \sum_{\xi_i \in \Xi_k} s(b_k, \xi_i), \tag{3.5}$$

and $S_k(b_k, \Xi_k)^2$ is the sample variance of $s(b_k, \xi_i)$ in Ξ_k, i.e.,

$$S_k(b_k, \Xi_k)^2 = \frac{\sum_{\xi_i \in \Xi_k} \left(s(b_k, \xi_i) - \bar{s}_k(b_k, \Xi_k)\right)^2}{N-1}. \tag{3.6}$$

For simplicity, we assume the batch size $|\Xi_k| = N$ is constant across all optimization iterations.

The evaluation of $\tilde{f}$ is expensive because, for each of the N realizations of ω, the acoustic efficiency s given in (3.2) requires the solution of a differential equation using a finite element method that involves the solution of a linear system of order $\mathcal{O}(30{,}000)$. (Ng and Willcox [29] employ a multifidelity approach to improve the efficiency of the sampling mechanism, but we will not consider it as it is not central to this investigation.)

3.3. Illustration. To visualize the behavior of the noisy function (3.4), we plot it in Figure 2 over a two-dimensional slice of $\mathbb{R}^6$ defined by varying two variables: b_3, b_4. The noise displays a discernible pattern rather than being highly erratic. As a result, the optimization problem is tractable notwithstanding the inherent nonlinearity of the simulation,

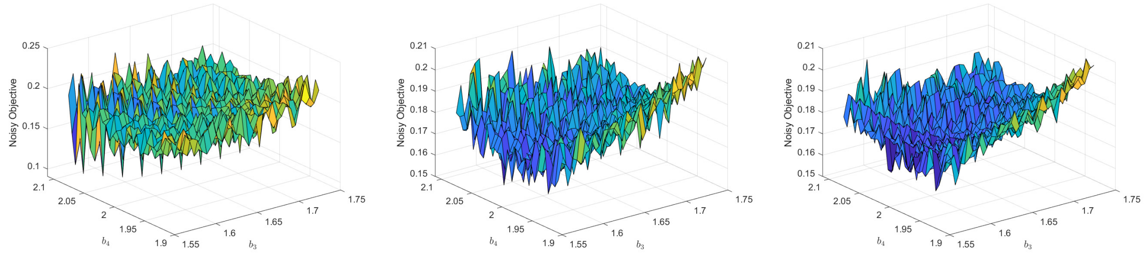

Figure 2: Noisy Function. The vertical axis plots the noisy objective (3.4) with different numbers of sample points: $N = 10$ (left), $N = 50$ (middle), and $N = 100$ (right). The horizontal axes represent values of two of the design variables, b_3 and b_4. Different realizations of the random variable ω were employed for each evaluation of $\tilde{f}$ in the region of interest.

The noise level $\sigma_f(b)$ in this function is defined as the standard deviation of $\tilde{f}(b)$ (see (2.3)), since the problem exhibits stochastic noise. In Figure 3, we plot an estimate $\epsilon_f(b)$ of $\sigma_f(b)$ (defined in (2.5) with $m = 50$) as we vary the variables b_3, b_4. While $\epsilon_f(b)$ does vary among different values of b, its fluctuations are not substantial. Thus, a single estimate might suffice for the optimization, as discussed in sections below.

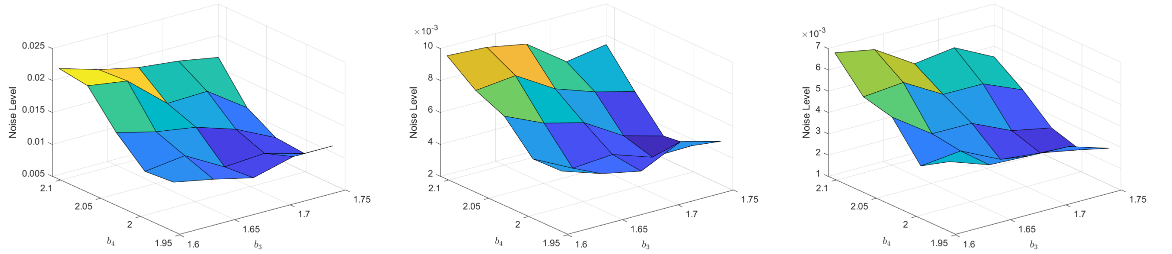

Figure 3: Noise Level. The vertical axis plots the estimated noise level $\epsilon_f(b)$ of the objective (3.4) with different numbers of sample points: $N = 10$ (left), $N = 50$ (middle), and $N = 100$ (right). The horizontal axes represent values of two of the design variables b_3 and b_4. Each value $\epsilon_f(b)$ is computed as defined in (2.5) with $m = 50$.

3.4. A Variant Illustrating Computational Noise.

The acoustic horn design problem can also be used to illustrate computational noise. As already mentioned, the finite element solution of the Helmholtz equation (3.1) requires solving of a non-symmetric linear system of equations. In the examples given in the previous sections this was done using a direct linear solver. However, practical applications often benefit from approximating solutions with an iterative method. In our next experiment, we utilize the GMRES method, with tolerance of 10^{-6}, to solve the linear system.

In order to isolate the effect of computational noise, we generate and fix a particular realization of $\Xi_k \equiv \Xi$, of size $N = 10$, in the evaluation of the objective function (3.4). We plot the generated objective function in Figure 4. For comparison, we also plot the function using a direct linear solver. In this context, computational noise is notably smaller than the stochastic noise previously illustrated. Although increasing the linear solver's tolerance can amplify the noise level, for brevity, our experiments will concentrate solely on stochastic noise.

4. Line Search Gradient Projection Methods.

In this section, we consider algorithms for the solution of noisy bound constrained optimization problems, such as the acoustic design problem (3.3). Our starting point is a classical gradient projection method with a backtracking line search, designed to be stable with respect to the critical operations discussed in the introduction.

Suppose the problem is defined in $\mathbb{R}^n$. Let $g(x) := \nabla f(x)$, and let $\tilde{g}(x)$ denote its noisy approximation. As is common, we denote $g_k := g(x_k)$ and $\tilde{g}_k := \tilde{g}(x_k)$. Given a search direction $\tilde{p}_k$, a straightforward extension of the Armijo sufficient decrease condition [30] reads

$$\tilde{f}(x_k + \beta_k \tilde{p}_k) - \tilde{f}(x_k) \le c \beta_k \tilde{g}_k^T \tilde{p}_k, \quad c \in (0, 1].$$

This requires the comparison of noisy function evaluations (case 1 in §1) and can lead to poor performance or failure [2, 31, 39]. To see this, suppose e.g. that $\tilde{p}_k = -\tilde{g}_k$. Then, the right hand side is always negative, but due to the noisy nature of $\tilde{f}$, the left hand side can be positive even for a very small steplength, forcing the line search to decrease β_k even more.

One approach for circumventing these difficulties is to introduce a margin $\epsilon_A(x_k)$ and to

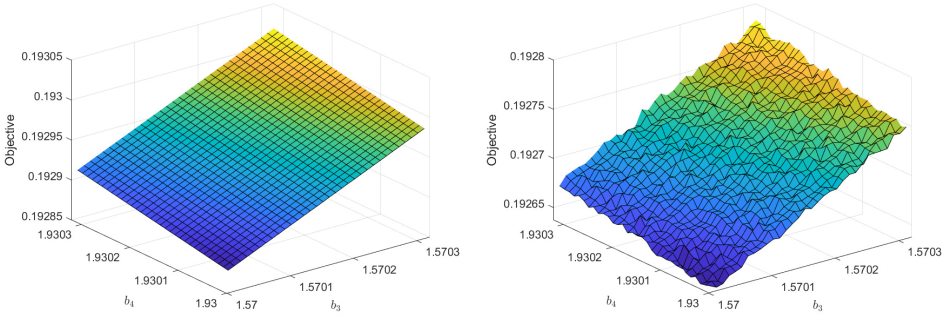

Figure 4: Computational Noise. The vertical axis plots a deterministic variant of the function (3.4) in which the samples have been fixed. The linear system within the PDE scheme is solved using a direct method (left) and using the iterative method GMRES with tolerance 10^{-6} (right).

relax the Armijo condition as follows [2, 3, 37, 39],

$$\tilde{f}(x_k + \beta_k \tilde{p}_k) \leq \tilde{f}(x_k) + c\beta_k \tilde{g}_k^T \tilde{p}_k + 2\epsilon_A(x_k). \tag{4.1}$$

A gradient projection method using a relaxed line search is given in Algorithm 1. It depends on a parameter α_0 that determines the initial trial point in the line search. The importance of α_0 will be discussed in subsequent sections. In the algorithm, $P_\Omega[\cdot]$ denotes the projection operator onto the feasible region Ω. For the moment, we assume that $\epsilon_A(x_k)$ depends on $\epsilon_f(x_k)$, and will elaborate on the exact nature of this relationship in the next subsection.

Algorithm 1: (GP-LS) Line Search Gradient Projection Method

```
Input: Initial point x_0, constants ρ ∈ (0,1), c ∈ (0,1), and initial trial steplength
   α_0 > 0.
Set k ← 0.
while a termination condition is not met do
    Determine ε_A(x_k).
    Compute a stochastic gradient g̃_k.
    p̃_k ← P_Ω[x_k − α_0 g̃_k] − x_k.
    Set β_k ← 1.
    while f̃(x_k + β_k p̃_k) > f̃(x_k) + c β_k g̃_k^T p̃_k + 2ε_A(x_k) do
        β_k ← ρ β_k.
    end
    x_{k+1} ← x_k + β_k p̃_k.
    Set k ← k + 1.
end
```

In our experiments we use the parameters $\rho = 1/2$ and $c = 10^{-4}$. We could have considered a more sophisticated gradient projection method with a projected backtracking line search [7], but the numerical and theoretical results would not be significantly different.

We now discuss the unspecified aspects of Algorithm 1, namely the computations of the relaxation $\epsilon_A(x_k)$ and the noisy gradients $\tilde{g}_k$.

4.1. Choosing the Relaxation $\epsilon_A(x)$.

One option is to choose $\epsilon_A(x)$ to be greater than ϵ_b, where the latter is defined in (2.4) as a bound on the noise. Then (4.1) is satisfied for all sufficiently small β_k, and one can establish deterministic convergence results to a neighborhood of the solution [2, 31, 39]. However, in many applications, computing the bound ϵ_b is not feasible. Even when it is possible, choosing $\epsilon_A(x) > \epsilon_b$ tends to be excessively cautious and can degrade performance, as we will demonstrate in §5.

A more effective approach, in general, is to choose $\epsilon_A(x) \leftarrow \lambda\epsilon_f(x)$, where $\epsilon_f(x)$ is the estimated noise level at x and λ is a positive constant. This rule is justified as follows.

Suppose that the random variable $\Delta(x)$ is i.i.d. for all $x \in \Omega$, and that $\sigma_f(x)$ remains constant, so that computing $\epsilon_f(x)$ at a single x suffices. Then by utilizing concentration inequalities we can see that, $\mathbb{E}(\Delta(x)) + \lambda\epsilon_f(x)$ serves as a high-probability estimate of ϵ_b for λ large enough. Given that the critical operations discussed in this paper solely involve comparisons or differences of function values, the mean cancels out, justifying the rule $\epsilon_A(x) \leftarrow \lambda\epsilon_f(x)$.

This rule can also be motivated in the absence of the i.i.d assumption by introducing the weaker set of assumptions: $\mathbb{V}(\Delta(x)) \le \sigma^2$ and $\mathbb{E}(\Delta(x)) = 0$ [11]. In that case it is reasonable to set $\epsilon_A(x) \leftarrow \lambda\sigma$. Another line of research [22, 23] that also motivates the rule $\epsilon_A(x) \leftarrow \lambda\epsilon_f(x)$ assumes the existence of probabilistic bounds of $\|\Delta(x)\|$, and allows for $\mathbb{E}(\Delta(x)) \neq 0$.

When the noise level does not vary significantly within the region of interest, it is more efficient to compute a constant estimate ϵ_f (as discussed in §2.1) and fine tune the parameter $\lambda \in [1, 2]$ to the application at hand. We can then drop the dependency on x and write

$$\epsilon_A \leftarrow \lambda\epsilon_f. \tag{4.2}$$

In case the distribution of $\Delta(x)$ varies dramatically for different x, one may have to recompute ϵ_A during the course of the optimization or employ ϵ_b in lieu of a fixed value ϵ_A; see Appendix B for details.

4.2. Finite Difference Gradient Approximation.

The gradient of the objective function can be approximated using (noisy) finite differences. This involves the critical operation 2 mentioned in §1. To achieve stability, the function evaluations must be spread out appropriately to balance truncation error and noise.

Let us consider the case where a universal noise level estimate ϵ_f is available for all x. A value of h that minimizes mean squared error for the forward difference estimator

$$[\tilde{g}^{FD}(x)]_i := \frac{\tilde{f}(x + he_i) - \tilde{f}(x)}{h}, \quad i = 1, \ldots, n,$$

is given by [26]

$$h \approx 8^{1/4}\sqrt{\frac{\epsilon_f}{L}}, \tag{4.3}$$

where L is a bound on the second derivative of the objective function (or the Lipschitz constant of the gradient). (In this formula, ϵ_f should be replaced by ϵ_b when the latter is the only information available.) Traditionally, the value of L is estimated independently from ϵ_f [17, 26, 36]. However, Shi et al. [38] recently introduced a bisection procedure that calculates h directly using only noisy evaluations $\tilde{f}$, avoiding a separate estimation of L.

In certain applications, such as the acoustic design problem described in §3, analytic expressions for the gradient of a sample average approximation of the objective function are available; see Appendix A. This will allow us to present a comparative efficiency analysis of noisy finite difference methods versus analytic gradients.

5. Numerical Experiments.

We now describe numerical experiments that test the efficiency of algorithms for solving noisy bound constrained optimization problem under various noise regimes. We compare the line search gradient projection method `GP-LS` defined in Algorithm 1 with a variant using a fixed steplength, referred to as `GP-F`, given by

$$x_{k+1} \leftarrow P_\Omega[x_k - \alpha \tilde{g}_k], \tag{5.1}$$

where $\tilde{g}_k$ is a gradient approximation, $P_\Omega[\cdot]$ is the projection operator onto the feasible region, and α is a fixed steplength determined at the start of the algorithm.

Unless otherwise noted, the algorithms tested in this paper operate in the sample inconsistent case, meaning that every evaluation of the function uses a different batch of samples. This applies both to finite difference approximations of gradients and to line searches. (As a benchmark, we report the results for the sample consistent case in Appendix C.)

5.1. Relaxed Line Search vs. Fixed Step Lengths.

It is common practice to avoid line searches when minimizing noisy functions. We investigate whether this practice is still justified when employing the relaxed line search (4.1). To do so, we test our acoustic design problem under increasing noise levels.

In the first set of experiments, we compare the two gradient projection algorithms, `GP-F` and `GP-LS`, using gradients generated by finite differences. We chose a sample size $N = 100$ in (3.4) for which the estimated noise level $\epsilon_f(b)$ varies between 10^{-3} and 10^{-2} (see Figure 3). Since $\epsilon_f(b)$ does not change dramatically, we use a single value ϵ_f. We set $\epsilon_A = 10^{-3}$ through (4.2), after experimenting with the value of λ. Similarly, we use a fixed finite difference interval $h = 10^{-2}$ in both methods, based on formula (4.3) (experiments for other values of h are discussed in the next subsection).

The results are displayed in Figure 5. Algorithm `GP-F` was tested using three values of the fixed steplength, $\alpha = 10^{-1}, 10^{-2}, 10^{-3}$. Algorithm `GP-LS` used an initial trial steplength $\alpha_0 = 1$. In the vertical axis we plot an approximation of the true objective function obtained by setting $N = 100$ in (3.4). In the left panel, the horizontal axis plots the iteration number; and in the right panel, it plots computational effort, defined as

$$N \times \text{number of function calls}. \tag{5.2}$$

We observe from Figure 5 that the performance of `GP-F` varies greatly with the choice of steplength α. The value $\alpha = 10^{-3}$ leads to a slow method, whereas the choice $\alpha = 10^{-1}$ results in wild oscillations. The best performing method, using $\alpha = 10^{-2}$, was identified after

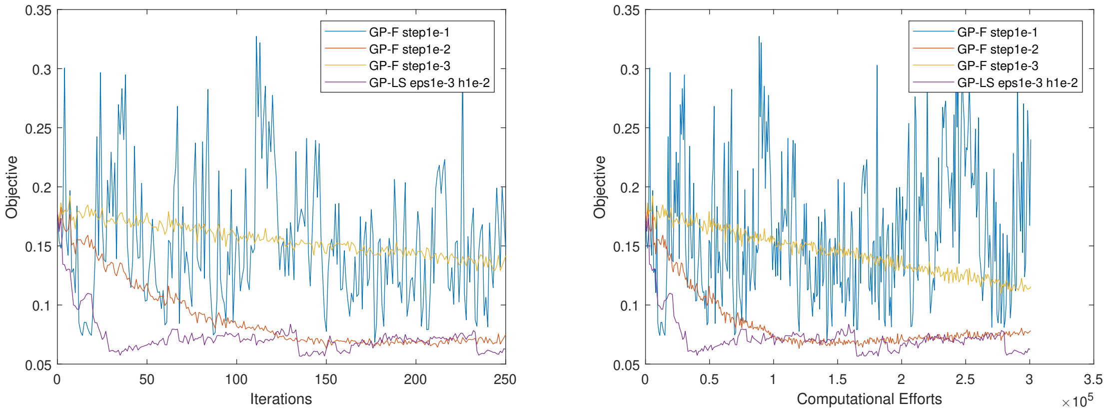

Figure 5: Comparison of the gradient projection method with (`GP-LS`) and without (`GP-F`) a line search; the former using a relaxation $\epsilon_A = 10^{-3}$ and the latter using three values of α. All methods use $N = 100$ and a finite difference interval $h = 10^{-2}$. Left: Objective function value vs. iteration. Right: Objective function value vs. computational effort.

extensive experimentation. Observe that `GP-LS` outperforms the best option of `GP-F` in the initial third of the run.

In the second set of experiments, we measure the effect of the relaxation parameter ϵ_A on algorithm `GP-LS`. Figure 6 reports results for choices $\epsilon_A = 10^{-2}, 10^{-3}, 10^{-4}$, which were derived as follows. For $N = 100$, letting $\lambda = 2$ and defining ϵ_f by (2.7), we have that $\epsilon_A \approx 10^{-2}$ (such estimate is close to ϵ_b). To seek a lower bound of $\epsilon_f(b)$, we set $\lambda = 1$, compute ϵ_f by randomly sampling $b_1, \dots, b_{100}$ in Ω, and let $\epsilon_f = \min_{i=1,\dots,100} \epsilon_f(b_i)$; this gives $\epsilon_A \approx 10^{-3}$. (We experiment with $\epsilon_A = 10^{-4}$ in order to observe the effect of underestimating ϵ_A.)

We observe from Figure 6 that `GP-LS` performs well for $\epsilon_A = 10^{-3}$ and 10^{-4} but not so for $\epsilon_A = 10^{-2}$. By using this upper bound, the algorithm accepts overly noisy steps, resulting in oscillations. In contrast, if the relaxation ϵ_A is chosen too small (i.e., 10^{-4}), it may cause the algorithm to repeatedly reject steps once it reaches the attainable accuracy in the function (observe the straight line in the right panel). However, this is not really harmful and a high number of rejections can be avoided by imposing a maximum number of backtracks; see e.g. the strategy in §5.5. In summary, it is advisable to choose ϵ_A to be in the lower range of the estimated values of $\epsilon_f(b)$.

5.2. Finite Differences vs. Analytic Gradients. A common view in optimization is that finite difference gradient approximations should be avoided in the noisy setting. We investigate this perspective in the context of the acoustic horn problem by comparing the use of finite differences and analytic expressions for the gradient of a sample average approximation of the function. These analytic expressions are provided by the PDE solver as discussed in Appendix A.

In Figure 7, we report the performance of the line search algorithm `GP-LS` using finite differences or analytic gradients. We set $N = 100$ and $\epsilon_A = 10^{-3}$ and obtain the estimate

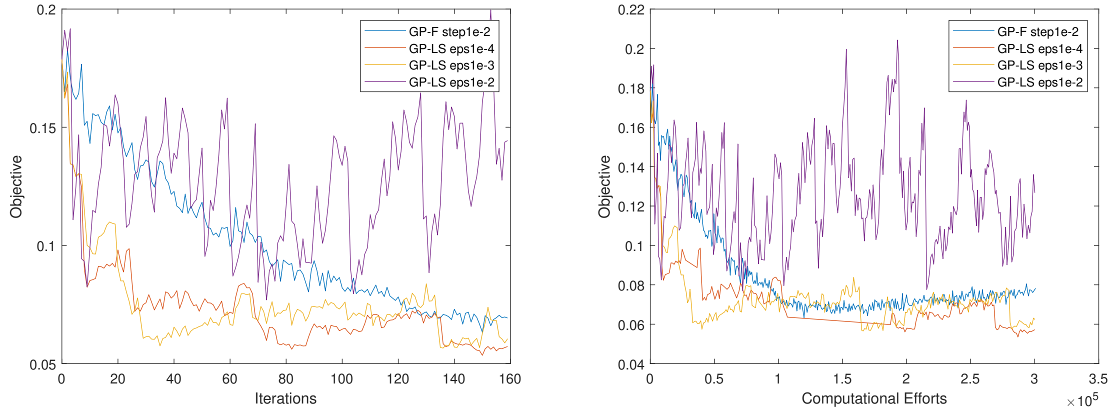

Figure 6: Performance of Algorithm GP-LS with three values (10^{-2}, 10^{-3}, 10^{-4}) of the relaxation parameter ϵ_A in the line search. We also plot the performance of Algorithm GP-F with $\alpha = 10^{-2}$. Left: Objective function value vs. iteration. Right: Objective function vs. computational effort (5.2).

$h \approx 10^{-2}$ by using formula (4.3) with $\epsilon_f = 10^{-3}$. Consequently, we report results with three values of the finite difference interval, namely $h = 10^{-1}, 10^{-2}, 10^{-3}$, to compare the outcomes of overestimating and underestimating interval choices. In the figure on the right we plot the objective function vs. CPU time, which is an appropriate measure since the cost of an analytic gradient evaluation is difficult to quantify in terms of function evaluations.

The plots in Figure 7 indicate that, as anticipated, the use of analytic gradients yields the best results. However, the margin of improvement is not significant compared to GP-LS with $h = 10^{-2}$, a value of h aligned with formula (4.3).

5.3. Increasing the Noise Level: $N = 50$. As the sample size decreases from $N = 100$ to $N = 50$, the problem becomes more noisy, potentially compromising the stability of the line search. Now, the convergence theory of stochastic gradient methods [8] states that the steplength should diminish in response to rising noise. This fact can be used to make the line search more robust by decreasing the initial trial steplength α_0 in GP-LS.

In Figure 8, we set $N = 50$, $\epsilon_A = 2 \times 10^{-3}$, and plot the results for GP-LS with $\alpha_0 = 1, 0.25, 10^{-3}$. While 0.25 is a reasonable choice, 1 and 10^{-3} are included to demonstrate the effects of excessively large or small choices of α_0. We also report the performance of GP-F with $\alpha = 10^{-2}$, a steplength obtained via tuning. The two figures report objective value vs. computational effort (defined in (5.2)). The left panel focuses on the early stage of the run while the right panel plots the overall long term behavior.

Figure 8 shows the benefits of using values of α_0 smaller than 1 in GP-LS. The choice $\alpha_0 = 0.25$ outperforms all other options including the tuned GP-F. The very small value $\alpha_0 = 10^{-3}$ leads to poor performance both because it limits the lengths of the steps unduly and because comparisons in the line search become unreliable, sometimes yielding repeated rejections of trial steplengths.

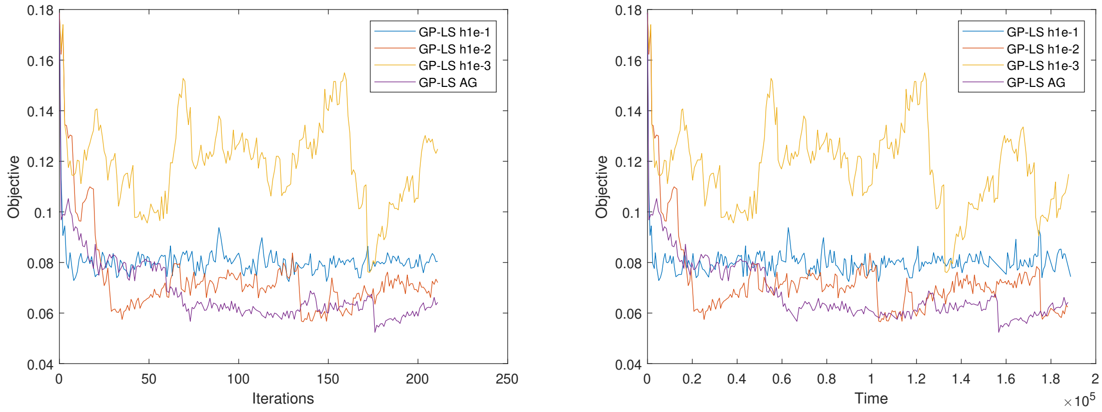

Figure 7: Comparison of analytic vs. finite difference gradients in Algorithm GP-LS. We report results for three values of the finite difference parameter $h = 10^{-1}, 10^{-2}, 10^{-3}$. Left: Objective function value vs. iteration. Right: Objective function value vs. CPU time.

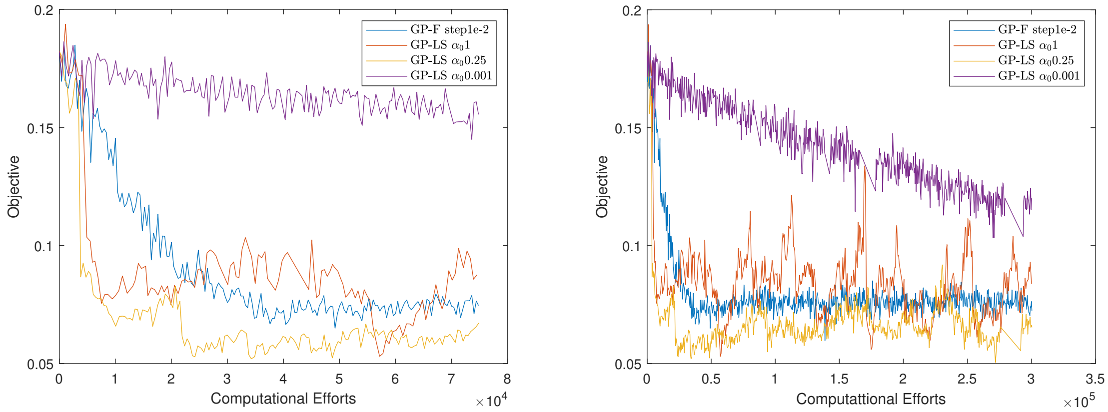

Figure 8: Comparison of three different values of the initial trial steplength, namely $\alpha_0 = 1, 0.25, 10^{-3}$ in GP-LS. We also report GP-F with $\alpha = 10^{-2}$. Both algorithms were tested using $N = 50$. Left: Objective function value vs. computational effort (up to $75,000$). Right: Objective function value vs. computational effort (up to 3×10^5).

5.4. A Higher Noise Level: $N = 10$. When $N = 10$, the noise level is so high that all algorithms exhibit strong oscillations in the objective. In Figure 9, we report results of GP-F with $\alpha = 10^{-2}$, and GP-LS with $\epsilon_A = 10^{-2}$ and $\alpha_0 = 0.025$ (all parameters chosen after experimentation). The panel on the left focuses on the initial stages of the run, and the right panel on the overall run. Note that the best objective value achievable by the methods is around 8×10^{-2}, whereas for $N = 50, 100$ it was 6×10^{-2}. GP-LS no longer has an advantage over GP-F, unlike the case for $N = 50$ or 100.

To summarize our experiments so far, the relaxed line search strategy performs efficiently in the presence of noise by reducing the initial trial point α_0 as the noise level increases. Yet, when dealing with highly noisy functions, employing a fixed step length strategy is equally effective. Nevertheless, we now demonstrate that further enhancements to the line search strategy are possible.

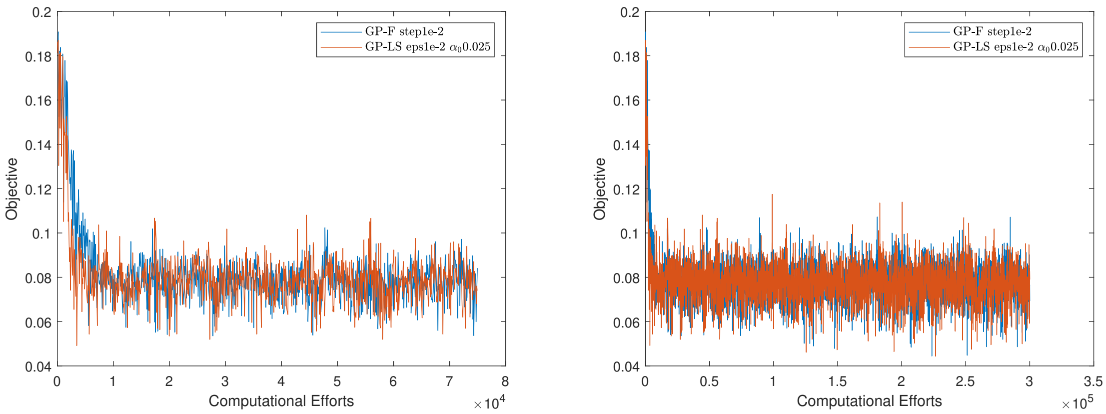

Figure 9: Comparison of GP-F and GP-LS with heuristics. Algorithm GP-LS and GP-F were tested using $N = 10$; Left: Objective function value vs. computational effort (up to $75,000$). Right: Objective function value vs. computational effort (up to 3×10^5).

5.5. A Self-Calibrated Line Search Strategy. We now show that the performance of the GP-LS method can be improved significantly in the highly noisy regime by adaptively selecting the two key parameters in the GP-LS method: ϵ_A and α_0. To do so, we first define a user-specified memory size T. Every T iterations, before computing the noisy gradients in GP-LS, instead of performing line 4 of Algorithm 1, we proceed as follows:

- Compute the average number of line search backtracks in the most recent T iterations, denoted as avg.
- If $avg \geq 3$, then update ϵ_A and α_0 as

$$\epsilon_A \leftarrow \min\{1.5\epsilon_A, 2\epsilon_f\}, \quad \alpha_0 \leftarrow \max\{0.5\alpha_0, 10^{-5}\}, \tag{5.3}$$

  and if $avg \leq 0.1$,

$$\epsilon_A \leftarrow \max\{0.5\epsilon_A, 10^{-5}\}, \quad \alpha_0 \leftarrow \min\{1.5\alpha_0, 10^{-1}\}. \tag{5.4}$$

The motivation for this strategy is as follows.

Case 1: If avg is large, then either the relaxation is too small and the line search has stagnated (see Figure 6 for $\epsilon_A = 10^{-4}$), or the search direction is too noisy leading to many backtracking steps. In this case, the strategy increases ϵ_A and decreases α_0 to further relax the line search and put more emphasis on safeguarding errors. The upper bound of ϵ_A is set as $2\epsilon_f$ since we have seen in §4.1 that line search will ultimately be successful with high probability as ϵ_A is increased to $2\epsilon_f$.

Case 2: If avg is small, then either ϵ_A is adequately large or the steps are productive. In this case, we decrease ϵ_A since we try to keep this parameter as small as possible, and increase α_0 to attempt to take more aggressive steps.

In addition to the rules (5.3) and (5.4), we limit the number of possible backtracks by requiring that β_k never be smaller than ρ^{3T}, where ρ is the contraction parameter defined in Algorithm 1. Thus, the condition in the while loop in line 8 of Algorithm 1 is changed to

$$\tilde{f}(x_k + \beta_k \tilde{p}_k) > \tilde{f}(x_k) + c\beta_k \tilde{g}_k^T \tilde{p}_k + 2\epsilon_A \text{ and } \beta_k \geq \rho^{3T}. \tag{5.5}$$

The constants in (5.3) and (5.4) can be tuned for the application at hand, but the method is not sensitive to the choices of these constants, with one caveat. It is important that, when changing ϵ_A and α_0, we decrease them more rapidly than increase them (note $1.5 \times 0.5 < 1$) because it is less harmful to perform more backtracks than accepting a poor step. We mention in passing that this method stands in contrast to a recently proposed method [40], where an estimation of the gradient norm variance was used to re-scale the steps.

The results of applying `GP-LS` with the self-calibrated strategy, denoted as `GP-LS-cal`, are displayed in Figure 10. There, $T = 5$ and the sample size is $N = 10$. The left panel compares fine-tuned `GP-LS` against `GP-LS-cal`. The right panel plots a smoothed version of the left figure, i.e., a moving average of objective. We can observe that `GP-LS-cal` clearly outperforms `GP-LS`. Moreover, the best average objective value of `GP-LS-cal` improves to around 6×10^{-2}, which is similar to the objective obtained for $N = 50$ and 100.

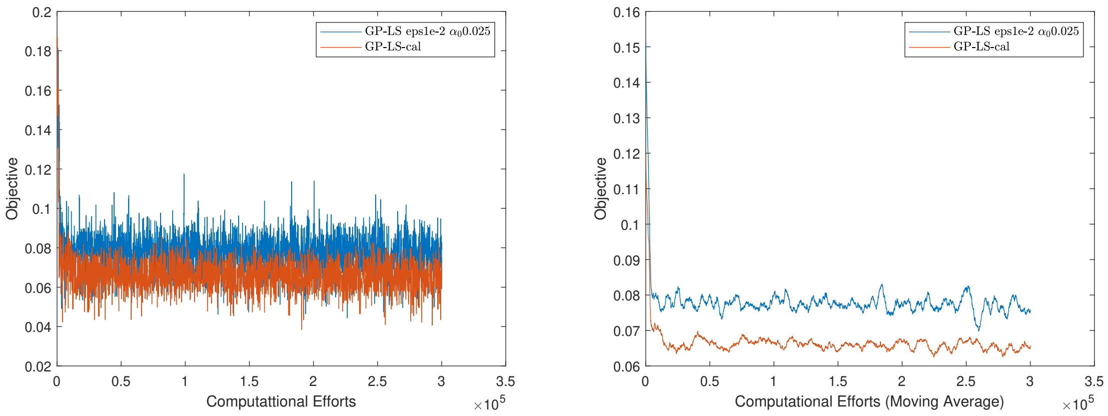

Figure 10: Comparison of `GP-LS` and `GP-LS-cal`. Algorithms were tested using $N = 10$, $T = 5$; Left: Objective function value vs. computational effort. Right: Moving average of recent 50 objective function values vs. computational effort.

6. Convergence Analysis.

In this section, we establish convergence properties for algorithm `GP-LS` in the presence of bounded noise, when applied to the problem

$$\min_{x \in \Omega} \quad f(x), \tag{6.1}$$

where f is a nonlinear function and Ω is a closed convex set. We begin by stating two common assumptions.

Assumption 6.1. *Ω is a nonempty, closed, and convex set, and for any $x \in \Omega$, $f(x) > -\infty$.*

Assumption 6.2. *f is continuously differentiable in the feasible region Ω and for all $x, y \in \Omega$, there exist $L > 0$ such that*

$$\|g(x) - g(y)\| \leq L\|x - y\|.$$

Next, we assume that the noise in the function and gradient is bounded.

Assumption 6.3. *For all $x \in \Omega$, there exists a constant $\epsilon_b > 0$ such that*

$$\|\tilde{f}(x) - f(x)\| \leq \epsilon_b.$$

Assumption 6.4. *For all $x \in \Omega$, there exists a constant $\epsilon_g > 0$ such that*

$$\|\tilde{g}(x) - g(x)\| \leq \epsilon_g.$$

Let us comment on the last two assumptions. In many engineering applications, including our acoustic design problem, the noise in the objective and gradient is inherently bounded due to the physical nature of the problem. In that case, when employing finite difference gradient estimations, we have that Assumption 6.2 and 6.3 imply Assumption 6.4. This justifies the results in [2, 3, 5, 31, 39] and the analysis given below, which assume bounded noise.

Nonetheless, a series of studies [9, 15, 22, 23] assume only probabilistic bounds on the noise, and as mentioned in §4.1, achieve high-probability convergence results. That analysis is more sophisticated but also more involved than the one presented here. Since we believe that the boundedness assumption holds in many applications, our analysis is relevant to practice. It is also novel in that no prior results exists for noisy gradient projection methods *with a line search*, to our knowledge.

We begin the proof of global convergence by citing several established lemmas and introducing a stationarity measure specifically tailored to this problem. Our ultimate objective is to demonstrate that the limit inferior of this measure is of order $O(\epsilon_b + \epsilon_g^2)$.

Lemma 6.1 (Prop. 1.1.9 Appendix B in [7]). *For any $x \in \mathbb{R}^n$, the projection of x on Ω exists and is unique. Furthermore, z is the projection of x on Ω if and only if $(x - z)^T(y - z) \leq 0$ for all $y \in \Omega$.*

Lemma 6.2 (Theorem 9.5-2 part (5) in [24]). *For any $x, y \in \mathbb{R}^n$,*

$$\|P_\Omega[x] - P_\Omega[y]\| \leq \|x - y\|.$$

We now recall a standard stationary measure from convex optimization [7]:

$$p(x) := P_\Omega[x - \alpha_0 g(x)] - x, \qquad \tilde{p}(x) := P_\Omega[x - \alpha_0 \tilde{g}(x)] - x, \tag{6.2}$$

where $\alpha_0 > 0$ is any initial step-length set in Algorithm 1. Note that by design of our gradient projection method, $p(x)$ is the search direction.

Lemma 6.3. *$x^* \in \Omega$ is a first-order stationary point of problem (6.1) if and only if $p(x^*) = 0$.*

This lemma is a simple extension of a classical result (see Prop. 6.1.1 (b) in [7]); we include its proof in Appendix D for completeness.

Remark 6.4. Lemma 6.3 implies that beyond serving as the search direction for algorithm GP-LS at iteration x_k, $p(x_k)$ also functions as a measure of stationary for problem (6.1). There is, however, another optimality measure that is more convenient in deriving our main convergence result. This measure is given by $-p(x_k)^T g(x_k)$, as discussed next.

Lemma 6.5. *$x^* \in \Omega$ is a first-order stationary point of problem (6.1) if and only if*

$$p(x^*)^T g(x^*) = 0. \tag{6.3}$$

Proof. By Lemma 6.3, it suffices to show that (6.3) is equivalent to $p(x^*) = 0$. Clearly $p(x^*) = 0 \;\Rightarrow\; p(x^*)^T g(x^*) = 0$.

To establish the converse, assume that $p(x^*)^T g(x^*) = 0$, and define θ as the angle between $p(x^*)$ and $g(x^*)$, so that

$$\|p(x^*)\| \|g(x^*)\| \cos\theta = 0.$$

If $\|p(x^*)\| = 0$ or $\|g(x^*)\| = 0$ (which by (6.2) implies $\|p(x^*)\| = 0$), then $p(x^*) = 0$, yielding the desired result.

Let us therefore consider the case when $\|p(x^*)\| \neq 0$ and $\|g(x^*)\| \neq 0$, and $\cos\theta = 0$. We show by contradiction that this case is not possible. Note from (6.2)

$$\begin{aligned}\|P_\Omega[x^* - \alpha_0 g(x^*)] - (x^* - \alpha_0 g(x^*))\|^2 &= \|P_\Omega[x^* - \alpha_0 g(x^*)] - x^*\|^2 + \|\alpha_0 g(x^*)\|^2 \\ &> \|P_\Omega[x^* - \alpha_0 g(x^*)] - x^*\|^2.\end{aligned}$$

This contradicts the fact that $P_\Omega[x^* - \alpha_0 g(x^*)]$ as the unique vector closest to $x^* - \alpha_0 g(x^*)$ in Ω. ∎

Using the standard abbreviations $p_k := p(x_k)$, $\tilde{p}_k := \tilde{p}(x_k)$, Lemma 6.5 establishes $p_k^T g_k$ as a stationary measure of problem (6.1)—and $\tilde{p}_k^T \tilde{g}_k$ is its noisy counterpart, which is the quantity accessed by the algorithm. In light of Lemma 6.1, it is easy to see that $-p_k^T g_k \geq 0$ and $-\tilde{p}_k^T \tilde{g}_k \geq 0$.

Let us now define

$$\delta_g(x_k) := (-\tilde{g}_k) - (-g_k), \qquad \delta_p(x_k) := \tilde{p}_k - p_k. \tag{6.4}$$

We now establish a technical lemma relating $-\tilde{p}_k^T \tilde{g}_k$ and the stationary measure $-p_k^T g_k$, in terms of a scaling factory dependent on the magnitude of the noise $\|\delta_g(x)\|$.

Lemma 6.6. *Under the assumptions previously stated, for any iterate x_k generated by* GP-LS *(Algorithm 1),*

$$-\tilde{p}_k^T \tilde{g}_k \geq \left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma_k^2\right)(-p_k^T g_k)$$

where

$$\gamma_k := \frac{\|\delta_g(x_k)\|}{\sqrt{-p_k^T g_k}}. \tag{6.5}$$

Proof. We lead the proof by noting the differences between $-p_k^T g_k$ and $-\tilde{p}_k^T \tilde{g}_k$:

$$-\tilde{p}_k^T \tilde{g}_k - (-p_k^T g_k) = -g_k^T \delta_p(x_k) + p_k^T \delta_g(x_k) + \delta_p(x_k)^T \delta_g(x_k). \tag{6.6}$$

We establish bounds on each terms on the right hand side of this equation.

We first show that the last term $\delta_p(x_k)^T \delta_g(x_k)$ is non-negative. Apply Lemma 6.1 with $x = x_k - \alpha_0 g_k$, $z = P_\Omega[x_k - \alpha_0 g_k] = x_k + p_k$, and $y = x_k + \tilde{p}_k$, we have $(-\alpha_0 g_k - p_k)^T(\tilde{p}_k - p_k) \leq 0$, which implies

$$-p_k^T \delta_p(x_k) \leq \alpha_0 g_k^T \delta_p(x_k). \tag{6.7}$$

Apply again Lemma 6.1 with $x = x_k - \alpha_0 \tilde{g}_k$, $z = P_\Omega[x_k - \alpha_0 \tilde{g}_k] = x_k + \tilde{p}_k$ and $y = x_k + p_k \in \Omega$, we have $(-\alpha_0 \tilde{g}_k - \tilde{p}_k)^T(p_k - \tilde{p}_k) \leq 0$, which implies

$$\tilde{p}_k^T \delta_p(x_k) \leq -\alpha_0 \tilde{g}_k^T \delta_p(x_k). \tag{6.8}$$

Summing up (6.7) and (6.8), we obtain

$$\begin{aligned} & (\tilde{p}_k - p_k)^T \delta_p(x_k) \leq \alpha_0 (g_k - \tilde{g}_k)^T \delta_p(x_k) \\ \Longrightarrow \quad & \delta_p(x_k)^T \delta_g(x_k) \geq \frac{1}{\alpha_0} \|\delta_p(x_k)\|^2 \geq 0. \end{aligned} \tag{6.9}$$

We next analyze the cross term $g_k^T \delta_p(x_k)$. For this, we first derive a few auxilary inequalities. First note by Lemma 6.2,

$$\|\delta_p(x_k)\| = \|p_k - \tilde{p}_k\| \leq \alpha_0 \|g_k - \tilde{g}_k\| = \alpha_0 \|\delta_g(x_k)\|. \tag{6.10}$$

Moreover, apply Lemma 6.1 with $x = x_k - \alpha_0 g_k$, $z = P_\Omega[x_k - \alpha_0 g_k] = x_k + p_k$, and $y = x_k \in \Omega$, we obtain

$$\|p_k\|^2 \leq -\alpha_0 p_k^T g_k. \tag{6.11}$$

To bound $g_k^T \delta_p(x_k)$, we have from (6.8)

$$\tilde{p}_k^T \delta_p(x_k) \leq -\alpha_0 \tilde{g}_k^T \delta_p(x_k) = -\alpha_0 g_k^T \delta_p(x_k) + \alpha_0 \delta_p(x_k)^T \delta_g(x_k). \tag{6.12}$$

Re-organize and obtain

$$\begin{aligned} -\alpha_0 g_k^T \delta_p(x_k) &\geq \tilde{p}_k^T \delta_p(x_k) - \alpha_0 \delta_p(x_k)^T \delta_g(x_k) \\ &= p_k^T \delta_p(x_k) + \delta_p(x_k)^T \delta_p(x_k) - \alpha_0 \delta_p(x_k)^T \delta_g(x_k) \\ &\geq -\alpha_0 \|\delta_g(x_k)\| \|p_k\| - \alpha_0^2 \|\delta_g(x_k)\|^2 \\ &\geq -\alpha_0 \|\delta_g(x_k)\| \sqrt{-\alpha_0 p_k^T g_k} - \alpha_0^2 \|\delta_g(x_k)\|^2 \\ &\geq \frac{\alpha_0}{2} p_k^T g_k - \frac{3\alpha_0^2}{2} \|\delta_g(x_k)\|^2 \\ &= \left(\frac{\alpha_0}{2} + \frac{3\alpha_0^2}{2} \gamma_k^2 \right) p_k^T g_k. \end{aligned} \tag{6.13}$$

Here, the second inequality follows from Cauchy-Schwartz inequality, $\|\delta_p(x_k)\|^2 \geq 0$, and (6.10); the third inequality follows from (6.11); the fourth is from arithmetic-geometric mean, i.e., $\alpha_0\|\delta_g(x_k)\|\sqrt{-\alpha_0 p_k^T g_k} \leq \frac{1}{2}\left(\alpha_0^2\|\delta_g(x_k)\|^2 - \alpha_0 p_k^T g_k\right)$; and the last line is by γ_k defined in (6.5).

Finally, using (6.13), (6.9) & Cauchy-Schwartz, (6.11), and arithmetic-geometric mean for the following inequalities respectively, we obtain the desired result:

$$
\begin{aligned}
-\tilde{p}_k^T\tilde{g}_k &= -p_k^T g_k + (-g_k^T\delta_p(x_k)) + p_k^T\delta_g(x_k) + \delta_p(x_k)^T\delta_g(x_k) \\
&\geq -p_k^T g_k + \left(\frac{1}{2} + \frac{3\alpha_0}{2}\gamma_k^2\right) p_k^T g_k + p_k^T\delta_g(x_k) + \delta_p(x_k)^T\delta_g(x_k) \\
&\geq \left(\frac{1}{2} - \frac{3\alpha_0}{2}\gamma_k^2\right)(-p_k^T g_k) - \|p_k\|\|\delta_g(x_k)\| \\
&\geq \left(\frac{1}{2} - \frac{3\alpha_0}{2}\gamma_k^2\right)(-p_k^T g_k) - \|\delta_g(x_k)\|\sqrt{-\alpha_0 p_k^T g_k} \\
&\geq \left(\frac{1}{2} - \frac{3\alpha_0}{2}\gamma_k^2\right)(-p_k^T g_k) - \frac{1}{2}\|\delta_g(x_k)\|^2 + \frac{\alpha_0}{2}p_k^T g_k \\
&= \left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma_k^2\right)(-p_k^T g_k). \qquad (6.14)
\end{aligned}
$$
■

We can now state the main convergence theorem for the gradient projection algorithm with a relaxed line search. We recall that $-p_k^T g_k$ serves both an algorithmic role in the Armijo decrease condition and a theoretical role as a stationary measure of the problem, as mentioned in Remark 6.4.

Theorem 6.7. *Under Assumptions 6.1-6.4, if $\alpha_0 + 2c < 1$ and $\epsilon_A > \epsilon_b$, the iterates $\{x_k\}$ generated by* `GP-LS` *(Algorithm 1) satisfy*

$$\liminf_{k\to\infty} \left|p_k^T g_k\right| \leq \bar{\epsilon}$$

where

$$\bar{\epsilon} := \frac{\epsilon_g^2}{\gamma^2} + \frac{2\alpha_0 L}{c\rho\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)}\left(\epsilon_A + \epsilon_b\right), \qquad (6.15)$$

and

$$\gamma^2 := \frac{(1-2c-\alpha_0)(1-\alpha_0)}{(1-2c-\alpha_0)(3\alpha_0+1)+2}. \qquad (6.16)$$

Proof. The proof is constructed by characterizing the descent on the objective function using the noisy stationary measure $\tilde{p}_k^T\tilde{g}_k$, and dividing the proof into two cases according to the relative size of the noise.

First, by applying Lemma 6.1 with $x = x_k - \alpha_0\tilde{g}_k$, $z = P_\Omega[x_k - \alpha_0\tilde{g}_k] = x_k + \tilde{p}_k$, and $y = x_k \in \Omega$, we have

$$\|\tilde{p}_k\|^2 \leq -\alpha_0\tilde{p}_k^T\tilde{g}_k. \qquad (6.17)$$

Next, by a Taylor expansion and Assumption 6.2, we have for any $\beta > 0$,

$$
\begin{aligned}
f(x_k + \beta \tilde{p}_k) &\le f(x_k) + \beta \tilde{p}_k^T g_k + \frac{L}{2}\beta^2 \|\tilde{p}_k\|^2 \\
&\le f(x_k) + \beta \tilde{p}_k^T (\tilde{g}_k + \delta_g(x_k)) + \frac{L}{2}\beta^2 (-\alpha_0 \tilde{p}_k^T \tilde{g}_k) \\
&\le f(x_k) + (-\beta + \frac{\alpha_0 L}{2}\beta^2)(-\tilde{p}_k^T \tilde{g}_k) + \beta \|\tilde{p}_k\| \|\delta_g(x_k)\| \\
&\le f(x_k) + (-\beta + \frac{\alpha_0 L}{2}\beta^2)(-\tilde{p}_k^T \tilde{g}_k) + \beta \|\delta_g(x_k)\| \sqrt{-\alpha_0 \tilde{p}_k^T \tilde{g}_k} \\
&\le f(x_k) + \left(\left(\frac{\alpha_0}{2} - 1 \right) \beta + \frac{\alpha_0 L}{2}\beta^2 \right) (-\tilde{p}_k^T \tilde{g}_k) + \frac{\beta}{2} \|\delta_g(x_k)\|^2,
\end{aligned} \tag{6.18}
$$

where the second and fourth inequalities are from (6.17), the third is from Cauchy-Schwartz, and the last is from the arithmetic-geometric mean. Together with Assumption 6.3, we have

$$
\tilde{f}(x_k + \beta \tilde{p}_k) \le \tilde{f}(x_k) + \left[\left(\left(\frac{\alpha_0}{2} - 1 \right) \beta + \frac{\alpha_0 L}{2}\beta^2 \right) (-\tilde{p}_k^T \tilde{g}_k) + \frac{\beta}{2} \|\delta_g(x_k)\|^2 \right] + 2\epsilon_b. \tag{6.19}
$$

We now note that the line search in `GP-LS` always terminates within finitely many backtracking steps. This follows from the fact that we pick $\epsilon_A > \epsilon_b$ and that the term inside square brackets in (6.19) converges to zero as $\beta \to 0$. Hence, the relaxed Armijo condition (4.1) will be satisfied for some sufficiently small $\beta_k > 0$.

We now divide the set of iterates into two cases depending on whether the noise dominates the optimality measure, in the sense that the ration γ_k is larger than the threshold γ, where these quantities are defined in (6.5) and (6.16), respectively.

Note by the assumption $\alpha_0 + 2c < 1$ and simple algebra

$$
0 < \gamma^2 < \frac{1 - \alpha_0}{3\alpha_0 + 1}. \tag{6.20}
$$

Case 1: Noise is relatively small: $\gamma_k^2 \le \gamma^2$. By (6.19), (6.5), and Lemma 6.6, we have

(6.21)

$$
\begin{aligned}
\tilde{f}(x_k + \beta \tilde{p}_k) &\le \tilde{f}(x_k) + \left(\left(\frac{\alpha_0}{2} - 1 \right) \beta + \frac{\alpha_0 L}{2}\beta^2 \right) (-\tilde{p}_k^T \tilde{g}_k) + \frac{\beta}{2} \gamma_k^2 (-p_k^T g_k) + 2\epsilon_b \\
&\le \tilde{f}(x_k) + \left(\left(\frac{\gamma_k^2}{1 - \alpha_0 - (3\alpha_0 + 1)\gamma_k^2} + \frac{\alpha_0}{2} - 1 \right) \beta + \frac{\alpha_0 L}{2}\beta^2 \right) (-\tilde{p}_k^T \tilde{g}_k) + 2\epsilon_b.
\end{aligned}
$$

With this result, the Armijo condition (4.1) holds when

$$
\left(\frac{\gamma_k^2}{1 - \alpha_0 - (3\alpha_0 + 1)\gamma_k^2} + \left(\frac{\alpha_0}{2} - 1 \right) \right) \beta + \frac{\alpha_0 L}{2}\beta^2 \le -c\beta, \tag{6.22}
$$

which is equivalent to

$$
\beta \le \frac{2}{\alpha_0 L} \left(-c - \frac{\gamma_k^2}{1 - \alpha_0 - (3\alpha_0 + 1)\gamma_k^2} - \frac{\alpha_0}{2} + 1 \right). \tag{6.23}
$$

Since $\gamma_k^2 \le \gamma^2$, by (6.20),

$$1-\alpha_0-(3\alpha_0+1)\gamma_k^2 \ge 1-\alpha_0-(3\alpha_0+1)\gamma^2 > 0. \tag{6.24}$$

With this, we note that $-\frac{\gamma_k^2}{1-\alpha_0-(3\alpha_0+1)\gamma_k^2}$ is decreasing in γ_k^2, for $\gamma_k^2 \in (0,\gamma^2]$. Therefore its lower bound is achieved when $\gamma_k=\gamma$, i.e.

$$-c-\frac{\gamma_k^2}{1-\alpha_0-(3\alpha_0+1)\gamma_k^2}-\frac{\alpha_0}{2}+1 \ge -c-\frac{\gamma^2}{1-\alpha_0-(3\alpha_0+1)\gamma^2}-\frac{\alpha_0}{2}+1=\frac{1}{2} \tag{6.25}$$

where the equality follows from the definition of γ^2 in (6.16) and algebra.

This, together with (6.23), implies that the relaxed Armijo condition holds for any $\beta \le \frac{1}{\alpha_0 L}$. Thus, for any $k \in \mathbb{N}$ in Case 1,

$$\beta_k \ge \frac{\rho}{\alpha_0 L}. \tag{6.26}$$

Therefore, we have from (4.1), Assumption 6.3, and (6.26) that

$$\begin{aligned} f(x_k+\beta_k\tilde{p}_k) &\le f(x_k)+c\beta_k\tilde{p}_k^T\tilde{g}_k+2\epsilon_A+2\epsilon_b \\ &\le f(x_k)-\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k)+2\epsilon_A+2\epsilon_b, \end{aligned} \tag{6.27}$$

which measures the reduction of the objective between any two consecutive iterations for Case 1 (notice that according to Algorithm 1, $x_{k+1}=x_k+\beta_k\tilde{p}_k$).

Case 2: Noise is relatively large: $\gamma_k^2 > \gamma^2$. By definition of γ_k in (6.16)

$$\|\delta_g(x_k)\|^2 > \gamma^2(-p_k^Tg_k). \tag{6.28}$$

As explained in the paragraph after (6.19), there always exists $\beta_k > 0$ such that the relaxed Armijo condition (4.1) holds. This fact, together with Assumption 6.3, Assumption 6.4, and (6.28),

$$\begin{aligned} f(x_k+\beta_k\tilde{p}_k) \le& f(x_k)-c\beta_k(-\tilde{p}_k^T\tilde{g}_k)+2\epsilon_A+2\epsilon_b \\ \le& f(x_k)+2\epsilon_A+2\epsilon_b \\ =& f(x_k)-\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k) \\ &+\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k)+2\epsilon_A+2\epsilon_b \\ \le& f(x_k)-\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k) \\ &+\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)\frac{\|\delta_g(x_k)\|^2}{\gamma^2}+2\epsilon_A+2\epsilon_b \\ \le& f(x_k)-\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k) \\ &+\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)\frac{\epsilon_g^2}{\gamma^2}+2\epsilon_A+2\epsilon_b \\ =& f(x_k)-\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2}-\frac{\alpha_0}{2}-\left(\frac{3\alpha_0}{2}+\frac{1}{2}\right)\gamma^2\right)(-p_k^Tg_k)+\eta, \end{aligned} \tag{6.29}$$

where

$$\eta := \frac{c\rho}{\alpha_0 L}\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)\frac{\epsilon_g^2}{\gamma^2} + 2\epsilon_A + 2\epsilon_b. \tag{6.30}$$

Condition (6.29) measures the reduction of the objective between any two consecutive iterations for Case 2.

Now combine both Case 1 and 2, and since $\eta > 2\epsilon_A + 2\epsilon_b$, it follows that for all $k \in \mathbb{N}$,

$$f(x_{k+1}) \leq f(x_k) - \frac{c\rho}{\alpha_0 L}\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)(-p_k^T g_k) + \eta. \tag{6.31}$$

Finally, to prove that $\liminf_{k\to\infty} |p_k^T g_k| \leq \bar{\epsilon}$ where $\bar{\epsilon}$ is defined in (6.15), assume for contradiction that there exists $\epsilon_1 > \bar{\epsilon}$ such that $-p_k^T g_k \geq \epsilon_1$. Then for all $k \in \mathbb{N}$,

$$f(x_{k+1}) \leq f(x_k) - \left[\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)\epsilon_1 - \eta\right]. \tag{6.32}$$

This shows that for each iteration there is a decrease in f of at least

$$\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)\epsilon_1 - \eta.$$

We conclude by noting that this quantity is strictly positive as $\epsilon_1 > \bar{\epsilon}$ and that $\frac{c\rho}{\alpha_0 L}\left(\frac{1}{2} - \frac{\alpha_0}{2} - \left(\frac{3\alpha_0}{2} + \frac{1}{2}\right)\gamma^2\right)\bar{\epsilon} - \eta = 0$. Therefore $f(x_k) \to -\infty$ as $k \to \infty$, which is a contradiction to Assumption 6.1. In light of Lemma 6.1 $-p_k^T g_k \geq 0$, and thus $\liminf_{k\to\infty} |p_k^T g_k| \leq \bar{\epsilon}$. ■

7. Final Remarks.

We underscore how this research extends beyond prior studies aimed at mitigating roundoff errors in optimization. Nonlinear optimization packages [12, 16, 27, 41, 43] and textbooks [14, 18, 30] devote attention to this issue. Nonetheless, the strategies for handling errors are introduced as heuristics that are seldom documented or justified. More critically, they tend to focus solely on roundoff errors,[1] characterized by machine precision ϵ_M, which is a precisely specified quantity. There is a need for a more comprehensive understanding of this topic in which stabilization techniques follow clearly specified guidelines, and where noise exhibits a more complex behavior than roundoff. This paper attempts to be a step toward that goal.

In the future, it would be desirable to conduct similar studies, using practical applications, for more general constrained optimization problems. We believe that the ideas presented here extend to such a wider setting.

Acknowledgments. The authors are grateful to Richard Byrd, Figen Oztoprak, and Stefan Wild for valuable discussions regarding the subject matter of this paper.

REFERENCES

[1] A notable exception is Chapter 2 in Gill, Murray and Wright [18] which considers other sources of errors, but their study is far from exhaustive.

[1] S. Bellavia, G. Gurioli, B. Morini, and P. Toint, *The impact of noise on evaluation complexity: The deterministic trust-region case*, arXiv preprint arXiv:2104.02519, (2021).

[2] A. S. Berahas, R. H. Byrd, and J. Nocedal, *Derivative-free optimization of noisy functions via quasi-Newton methods*, SIAM Journal on Optimization, 29 (2019), pp. 965–993.

[3] A. S. Berahas, L. Cao, and K. Scheinberg, *Global convergence rate analysis of a generic line search algorithm with noise*, SIAM Journal on Optimization, 31 (2021), pp. 1489–1518.

[4] A. S. Berahas, F. E. Curtis, M. J. O'Neill, and D. P. Robinson, *A stochastic sequential quadratic optimization algorithm for nonlinear equality constrained optimization with rank-deficient Jacobians*, arXiv preprint arXiv:2106.13015, (2021).

[5] A. S. Berahas, F. E. Curtis, D. Robinson, and B. Zhou, *Sequential quadratic optimization for nonlinear equality constrained stochastic optimization*, SIAM Journal on Optimization, 31 (2021), pp. 1352–1379.

[6] D. P. Bertsekas, *Incremental gradient, subgradient, and proximal methods for convex optimization: A survey*, Optimization for Machine Learning, (2010), pp. 1–38.

[7] ———, *Convex Optimization Algorithms*, Athena Scientific, 2015.

[8] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, Siam Review, 60 (2018), pp. 223–311.

[9] L. Cao, A. S. Berahas, and K. Scheinberg, *First-and second-order high probability complexity bounds for trust-region methods with noisy oracles*, Mathematical Programming, (2023), pp. 1–52.

[10] C. Cartis, N. I. M. Gould, and P. Toint, *Strong evaluation complexity of an inexact trust-region algorithm with for arbitrary-order unconstrained nonconvex optimization*, arXiv preprint arXiv:2001.10802, (2021).

[11] R. Chen, M. Menickelly, and K. Scheinberg, *Stochastic optimization using a trust-region method and random models*, Mathematical Programming, 169 (2018), pp. 447–487.

[12] A. R. Conn, N. I. M. Gould, and P. L. Toint, LANCELOT*: a Fortran package for Large-scale Nonlinear Optimization (Release A)*, Springer Series in Computational Mathematics, Springer Verlag, Heidelberg, Berlin, New York, 1992.

[13] F. E. Curtis, K. Scheinberg, and R. Shi, *A stochastic trust region algorithm based on careful step normalization*, INFORMS Journal on Optimization, 1 (2019), pp. 200–220.

[14] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983. Reprinted as *Classics in Applied Mathematics 16*, SIAM, Philadelphia, Pennsylvania, USA, 1996.

[15] Y. Fang, S. Na, M. W. Mahoney, and M. Kolar, *Fully stochastic trust-region sequential quadratic programming for equality-constrained optimization problems*, arXiv preprint arXiv:2211.15943, (2022).

[16] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Review, 47 (2005), pp. 99–131.

[17] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Computing forward-difference intervals for numerical optimization*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 310–321.

[18] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[19] N. I. M. Gould and P. Toint, *An adaptive regularization algorithm for unconstrained optimization with inexact function and derivatives values*, arXiv preprint arXiv:2111.14098, (2021).

[20] S. Granville, *Optimal reactive dispatch through interior point methods*, IEEE Transactions on power systems, 9 (1994), pp. 136–146.

[21] R. W. Hamming, *Introduction to Applied Numerical Analysis*, Courier Corporation, 2012.

[22] B. Jin, K. Scheinberg, and M. Xie, *High probability complexity bounds for adaptive step search based on stochastic oracles*, arXiv preprint arXiv:2106.06454, (2021).

[23] ———, *Sample complexity analysis for adaptive optimization algorithms with stochastic oracles*, arXiv preprint arXiv:2303.06838, (2023).

[24] E. Kreyszig, *Introductory functional analysis with applications*, vol. 17, John Wiley & Sons, 1991.

[25] J. J. Moré and S. M. Wild, *Estimating computational noise*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1292–1314.

[26] ———, *Estimating derivatives of noisy simulations*, ACM Transactions on Mathematical Software (TOMS), 38 (2012), p. 19.

[27] B. A. Murtagh and M. A. Saunders, *MINOS 5.4 user's guide*, tech. rep., SOL 83-20R, Systems Optimization Laboratory, Stanford University, 1983. Revised 1995.

[28] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization, 19 (2009), pp. 1574–1609.

[29] L. W. Ng and K. E. Willcox, *Multifidelity approaches for optimization under uncertainty*, International Journal for numerical methods in Engineering, 100 (2014), pp. 746–772.

[30] J. Nocedal and S. Wright, *Numerical Optimization*, Springer New York, 2 ed., 1999.

[31] F. Oztoprak, R. Byrd, and J. Nocedal, *Constrained optimization in the presence of noise*, arXiv preprint arXiv:2110.04355, (2021).

[32] B. T. Polyak, *Introduction to optimization. optimization software*, Inc., Publications Division, New York, 1 (1987), p. 32.

[33] H. Robbins and S. Monro, *A stochastic approximation method*, The Annals of Mathematical Statistics, (1951), pp. 400–407.

[34] S. M. Robinson, *Perturbed kuhn-tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Mathematical programming, 7 (1974), pp. 1–16.

[35] J. G. Saw, M. C. Yang, and T. C. Mo, *Chebyshev inequality with estimated mean and variance*, The American Statistician, 38 (1984), pp. 130–132.

[36] H.-J. M. Shi, M. Qiming Xuan, F. Oztoprak, and J. Nocedal, *On the numerical performance of finite-difference-based methods for derivative-free optimization*, Optimization Methods and Software, (2022), pp. 1–23.

[37] H.-J. M. Shi, Y. Xie, R. Byrd, and J. Nocedal, *A noise-tolerant quasi-newton algorithm for unconstrained optimization*, SIAM Journal on Optimization, 32 (2022), pp. 29–55.

[38] H.-J. M. Shi, Y. Xie, M. Q. Xuan, and J. Nocedal, *Adaptive finite-difference interval estimation for noisy derivative-free optimization*, SIAM Journal on Scientific Computing, 44 (2022), pp. A2302–A2321.

[39] S. Sun and J. Nocedal, *A trust region method for the optimization of noisy functions*, arXiv preprint arXiv:2201.00973, (2022).

[40] S. Sun and Y. Xie, *Stochastic ratios tracking algorithm for large scale machine learning problems*, arXiv preprint arXiv:2305.09978, (2023).

[41] A. Wächter, *An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*, PhD thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2002.

[42] Y. Xie, R. H. Byrd, and J. Nocedal, *Analysis of the BFGS method with errors*, SIAM Journal on Optimization, 30 (2020), pp. 182–209.

[43] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *Algorithm 78: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization*, ACM Transactions on Mathematical Software, 23 (1997), pp. 550–560.

Appendix A. Analytical Gradient of the Design Problem.

In the acoustic horn design problem outlined in (3.3), the gradient of the sample approximation in (3.4) can be computed analytically. More specifically,

$$\tilde{g}_k \triangleq \nabla f(b_k, \Xi_k) = \nabla \bar{s}_k(b_k, \Xi_k) + 3\nabla \sqrt{S_k(b_k, \Xi_k)^2}, \tag{A.1}$$

where $\bar{s}_k(b_k, \Xi_k)$ and $S_k(b_k, \Xi_k)^2$ are defined in (3.5) and (3.6). Furthermore,

$$\nabla \bar{s}_k(b_k, \Xi_k) = \frac{1}{N} \sum_{\xi_i \in \Xi_k} \nabla s(b_k, \xi_i), \tag{A.2}$$

and simple algebra shows that,

$$\nabla \sqrt{S_k(b_k, \Xi_k)^2} = \frac{1}{N-1} \frac{\sum_{\xi_i \in \Xi_k} (s(b_k, \xi_i) - \bar{s}_k(b_k, \Xi_k))(\nabla s(b_k, \xi_i) - \nabla \bar{s}_k(b_k, \Xi_k))}{\sqrt{S_k^2(b_k, \Xi_k)}}. \tag{A.3}$$

If Ξ_k is randomly sampled, $\tilde{g}_k$ is an unbiased estimator for the gradient in (3.3) and can be computed by approximating $\nabla s(b_k, \xi_i)$ for each $\xi_i \in \Xi_k$. In the definition of s from (3.2), Γ_{inlet} is independent of b. If u is smooth, with $\mathbb{1}$ indicating $\int_{\Gamma_{\text{inlet}}} u d\Gamma \geq 0$,

$$\nabla s(b, \xi_i) = (2\mathbb{1} - 1) \int_{\Gamma_{\text{inlet}}} \nabla u d\Gamma. \tag{A.4}$$

Here ∇u can be obtained as a by-product while solving the Helmholtz equation with a finite element solver. Numerical integration over Γ_{inlet} yields $\nabla s(b, \xi_i)$ and $\tilde{g}_k$.

Appendix B. Further Discussion on Noise Level.

In §4.1, we justified the rule $\epsilon_A \leftarrow \lambda \epsilon_f$, under specific assumptions on $\Delta(x)$. However, these assumptions may not be valid in cases where the noise distribution varies significantly across different x values. This limitation is evident in scenarios where canceling the mean is not possible, as discussed in §4.1.

While the self-calibrated strategy proposed in §5.5– which can be viewed as an implicit way of estimating the local noise level– is one possible solution, it may fail to provide sufficient safeguards in some extreme cases (e.g. when the algorithm is highly sensitive to the choice of ϵ_A). For those scenarios, we still need to estimate a bound on the noise ϵ_b (defined in (2.4)) or a high-probability bound. We discuss such estimation next.

B.1. Estimation of ϵ_b for Stochastic Noise.

For simplicity, we will obtain ϵ_b by computing an estimate of $\sup \|\Delta(x)\|$ at a representative x. A global estimate can then be derived e.g. by (2.7).

One can establish consistent estimators of the noise bound if we can compute an estimate on the true objective value. Let us generate m i.i.d. samples $\{\tilde{f}_1(x), \tilde{f}_2(x), \ldots, \tilde{f}_m(x)\}$ and let us compute an accurate estimate of the true objective $f(x)$, denoted as $\widehat{f}(x)$. Then the samples of noise in the function are given by

$$\delta_j(x) := \tilde{f}_j(x) - \widehat{f}(x), \quad j = 1, 2, \cdots, m. \tag{B.1}$$

A concrete example arises in stochastic optimization where the true objective is $f(x) := \mathbb{E}(F(x,\xi))$. The jth sample of the noisy objective is defined as $\tilde{f}_j(x) = \frac{1}{N}\sum_{i=1}^{N} F(x,\xi_{j_i})$ for an i.i.d. batch $\{\xi_{j_1}, \xi_{j_2}, \cdots, \xi_{j_N}\}$ of size N. An accurate estimator $\widehat{f}(x)$ of $f(x)$ can then be defined as $\widehat{f}(x) = \frac{1}{M}\sum_{i=1}^{M} F(x,\xi_i)$ for another batch of i.i.d. samples $\{\xi_i\}_{i=1}^{M}$, where $M \gg N$ is sufficiently large.

We provide the following three estimators that can be used in practice, where the first two require the access to $\widehat{f}(x)$ and the third one does not:

1) *Empirical Chebyshev bound [35]:*

$$\hat{\epsilon}_b^1 := \overline{\delta(x)} + \lambda\sqrt{\frac{1}{m-1}\sum_{j=1}^{m}(\delta_j(x) - \overline{\delta(x)})^2} \tag{B.2}$$

for some integer λ large enough, where $\overline{\delta(x)} = [\delta_1(x) + \cdots + \delta_m(x)]/m$.

2) *Maximum of $|\delta_j(x)|$:*

$$\hat{\epsilon}_b^2 := \max_{j=1,\cdots,m}\{|\delta_j(x)|\}. \tag{B.3}$$

3) *Range of noisy objectives:*

$$\hat{\epsilon}_b^3 := \max_{j=1,\cdots,m} \tilde{f}_j(x) - \min_{j=1,\cdots,m} \tilde{f}_j(x). \tag{B.4}$$

$\hat{\epsilon}_b^1$ is a high-probability bound of $\|\Delta(x)\|$, assuming that the noise has a finite variance but not necessarily bounded. $\hat{\epsilon}_b^2$ is a consistent estimator of ϵ_b if $\sup\|\Delta(x)\| < \infty$. $\hat{\epsilon}_b^3$ can be a biased (and depending on the estimated quantity, potentially inconsistent) estimator if the noise does not have mean zero, yet it can be easily computed without $\widehat{f}(x)$. In practice, $\hat{\epsilon}_b^3$ is an attractive candidate when $\widehat{f}(x)$ is expensive or not accessible, or when the noise level estimate is not required to be accurate, as in the acoustic horn design.

B.2. Estimating ϵ_b for Computational Noise. Due to the deterministic nature of computational noise, the first two estimators discussed above cannot be employed. As an alternative, we can modify the range estimator (B.4) following a similar approach as `ECNoise`. At a selected point x, one can collect noisy objectives in a small neighborhood of x, and then compute the range as an estimate of ϵ_b. Similar to the argument for stochastic noise, if the distribution does not vary significantly, using `ECNoise` is usually effective; see [25].

Appendix C. Sample Selection and Consistency.

In many stochastic optimization problems, such as the acoustic horn design described in §3, the noisy evaluations $\tilde{f}(x_k)$, e.g. (3.4), depend on a particular sample batch Ξ_k. In certain cases, the selection of Ξ_k is entirely under the control of the user. One can thus fix Ξ_k during the course of an iteration of the optimization algorithm, a case we refer to as "sample consistency". In such a setting the effect of noise on function comparisons and differences is more benign.

Reusing samples is, however, not always possible. In that case, the algorithm will operate in the "sample inconsistent" regime, which is the most general and challenging for optimization methods and holds particular interest in this paper.

Let us summarize these two cases for the key components of our algorithm.

Relaxed line search. For backtrack numbers $\ell = 1, 2, \cdots$, we denote the sample used in the evaluation of $\tilde{f}(x_k + \beta_k^\ell \tilde{p}_k)$ by Ξ_k^ℓ. In the sample inconsistent case, the Ξ_k^ℓ are different from each other and a relaxation ϵ_A is employed. On the other hand, if sample consistency is ensured, we can set $\epsilon_A \leftarrow 0$ since no errors are involved in the comparison with a fixed Ξ_k [2].

Finite differences. Given the estimated noise level ϵ_f, the finite difference estimator is

$$\text{(C.1)} \qquad [\tilde{g}_k]_i := \frac{\tilde{f}(b_k + he_i, \Xi_k^2) - \tilde{f}(b_k, \Xi_k^1)}{h} \qquad i = 1, \cdots, n,$$

where Ξ_k^1 and Ξ_k^2 are two batches. Sample inconsistency allows $\Xi_k^1 \neq \Xi_k^2$, and h needs to be chosen according to the noise level as seen in (4.3). With sample consistency, $\Xi_k^1 = \Xi_k^2$, formula (C.1) gives a fairly accurate gradient approximation of the corresponding sample average approximation of the objective, and thus h is set as the unit roundoff ϵ_M.

C.1. Numerical Results with Sample Consistency. We study the performance of algorithm `GP-LS` when fixing the sample during line search and gradient estimation. In Figure 11, we plot the performance of `GP-LS` with $\epsilon_A = 0$, and for $N = 10, 50, 100$. For each value of N, we adjust α_0 $(0.1, 0.25, 1$ respectively$)$ to cope with the fact that the sample average approximations of the objective function become increasingly inaccurate as N decreases. The finite difference interval h is chosen to be 10^{-6} for all cases.

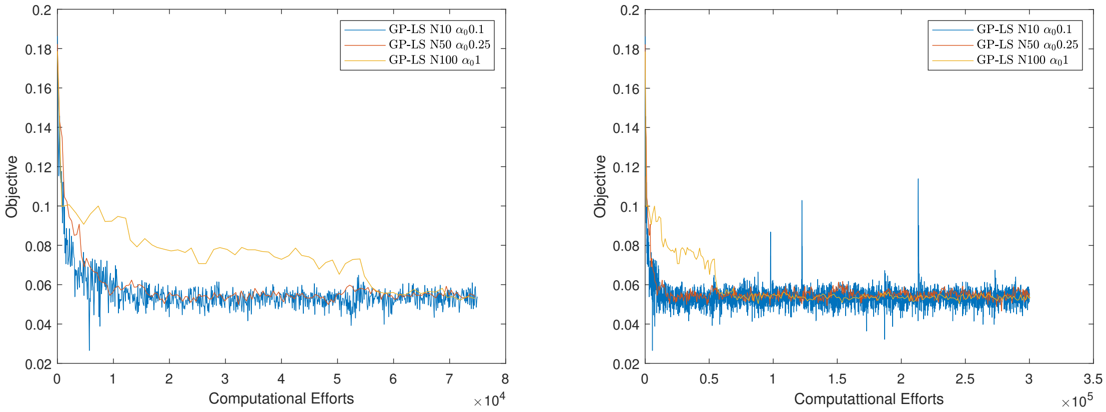

Figure 11: Comparison of different sample sizes when using a sample consistent version of Algorithm `GP-LS` using sample sizes $10, 50, 100$, and different α_0 respectively; Left: Objective function value vs. computational effort (up to $75,000$). Right: Objective function value vs. computational effort (up to 3×10^5).

We observe in Figure 11 that all three plots exhibit nice convergence behavior. With smaller sample sizes the iterates approach the solution more quickly, although they may give rise to spikes as the iteration continues. We conclude that, when feasible, sample consistency

[2]Note that although the comparison is robust, $\tilde{f}(x_k)$ is still a noisy estimate and a careful choice of α_0 can be useful when noise is large; see §C.1.

results in robust and efficient performance, if an appropriate value of the sample size N is first determined after experimentation.

Appendix D. Supplementary Proof.

Lemma D.1. *$x^* \in \Omega$ is a first-order stationary point of problem (6.1) if and only if $p(x^*) = 0$.*

Proof. Prop. 6.1.1 (b) in [7] shows ($\Leftarrow$) of Lemma 6.3.

To see ($\Rightarrow$), since x^* is a stationary point and by definition, $g(x^*)^T(x - x^*) \geq 0$ for all $x \in \Omega$. Take $x = P_\Omega[x^* - \alpha_0 g(x^*)]$, then

$$g(x^*)^T(P_\Omega[x^* - \alpha_0 g(x^*)] - x^*) = p(x^*)^T g(x^*) \geq 0. \tag{D.1}$$

Note that by letting $x = x^* - \alpha_0 g(x^*)$, $z = P_\Omega[x^* - \alpha_0 g(x^*)]$ and $y = x^*$ in Lemma 6.1, one has

$$\begin{aligned} &(x^* - \alpha_0 g(x^*) - P_\Omega[x^* - \alpha_0 g(x^*)])^T (x^* - P_\Omega[x^* - \alpha_0 g(x^*)]) \leq 0 \\ \Longrightarrow &\|p(x^*)\|^2 = \|x^* - P_\Omega[x^* - \alpha_0 g(x^*)]\|^2 \leq -\alpha_0 p(x^*)^T g(x^*) \leq 0 \end{aligned}$$

where the final inequality follows from (D.1). This implies that $p(x^*) = 0$. ■