A DERIVATIVE-FREE OPTIMIZATION ALGORITHM FOR THE EFFICIENT MINIMIZATION OF TIME-AVERAGED STATISTICS

POORIYA BEYHAGHI AND THOMAS BEWLEY

Abstract This paper considers the efficient minimization of the infinite time average of a stationary ergodic process in the space of a handful of design parameters which affect it. Problems of this class, derived from physical or numerical experiments which are sometimes expensive to perform, are ubiquitous in engineering applications. In such problems, any given function evaluation, determined with finite sampling, is associated with a quantifiable amount of uncertainty, which may be reduced via additional sampling. The present paper proposes a new optimization algorithm to adjust the amount of sampling associated with each function evaluation, making function evaluations more accurate (and, thus, more expensive), as required, as convergence is approached. The work builds on our algorithm for Delaunay-based Derivative-free Optimization via Global Surrogates (Δ-DOGS, see JOGO DOI: 10.1007/s10898-015-0384-2). The new algorithm, dubbed α-DOGS, substantially reduces the overall cost of the optimization process for problems of this important class. Further, under certain well-defined conditions, rigorous proof of convergence to the global minimum of the problem considered is established.

1. Introduction. In this paper, the Delaunay-based derivative-free optimization algorithm developed in [4–6], dubbed Δ-DOGS, is modified to minimize an objective function, $f(x) : \mathbb{R}^n \to \mathbb{R}$, obtained by taking the infinite-time average of a discrete-time ergodic process $g(x,k)$ for $k = 1, 2, 3, \ldots$ such that

$$f(x) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} g(x,k), \tag{1a}$$

where, for $k \gtrsim \bar{k}$, $g(x,k)$ is assumed to be statistically stationary. The feasible domain in which the optimal design parameter vector $x \in \mathbb{R}^n$ is sought is a bound constrained domain

$$L = \{x | a \leq x \leq b\} \quad \text{where } a < b \in \mathbb{R}^n. \tag{1b}$$

Note that calculating the true value of $f(x)$ for any x is impossible; $f(x)$ can only be approximated as the average of $g(x,k)$ over some *finite* number of samples N. The truth function $f(x)$ is typically a smooth function of x, though it is often nonconvex; computable approximations of $f(x)$, however, are generally *nonsmooth* in x, as the truncation error (associated with the fact any approximation of $f(x)$ must be computed with finite N) is effectively decorrelated from one approximation of $f(x)$ to the next.

Minimizing (1a) within the feasible domain (1b) is the subject of interest in host of practical applications, such as the optimization of stiffness and shape parameters and feedback control gains in mechanical systems and manufacturing processes involving turbulent flows, the setting of airline ticket prices, etc. Remarkably, however, there are very few optimization algorithms today ([13] and [26] being notable exceptions) that specifically address problems of this form, which instead are typically handled using classical derivative-free optimization approaches which, quite inefficiently, approximate each function evaluation with the same amount of sampling. Though many such classical derivative-free optimization approaches effectively keep function

evaluations far apart in parameter space until convergence is approached, thereby mitigating somewhat the effect of uncertainty in the function evaluations, at least for a while, they are not specifically designed to adjust the amount of sampling associated with each individual function evaluation, making function evalutations more accurate (and, thus, more expensive), as required, as convergence is approached. The algorithm presented here, in contrast, automatically adjusts the sampling associated with each function evaluation as convergence is approached. Three existing classes of algorithms that might be considered to optimize problems of the form given in (1) are discussed further below.

The first is based on a convex production planning model [16], using the dynamic programming principle to find an optimal solution. This modeling has been analyzed in both the discrete [11] and continuous [24, 25] settings. This approach is appropriate for control problems in which an adaptive controller is implemented with a large number of design parameters.

The second is based on an adjoint formulation (see, e.g., [10, 12, 20, 21]), which calculates the gradient of the function of interest at each iteration based on an adjoint computation. This method is very powerful for many problems in which essentially exact function evaluations are available, and scales exceptionally well to problems with many design parameters. However, for problems of the form (1), such as the optimization of airfoil shapes for turbulent flows, classical adjoint formulations are not suitable. This issue is addressed in a clever way in [32] and [31], where a novel method to alleviate it is presented. Nevertheless, this approach still only assures local convergence, and is highly sensitive to the accuracy of the mathematical model which is used for adjoint-based computations of the gradient.

The third is the class of derivative-free methods, which are indeed the most promising class of approaches for problems of the present form. These methods are also implemented for shape optimization in airfoil design [14], as well as in online optimization [13]. With such methods, only values of the function evaluations themselves are used, and neither a derivative nor its estimate is needed. The best methods of this class strive to keep function evaluations far apart in parameter space until convergence is approached, thereby mitigating somewhat the effect of uncertainty in the function evaluations. This class of methods generally handles feasibility boundaries quite well, and may be used to globally minimize the function of interest. However, this class of method scales quite poorly with dimension of the problem. The surrogate management framework [7] and Bayesian algorithms [18, 19, 23, 26] are amongst the best derivative-free methods available today, and are implemented for minimizing a problem of the form in (1) in [14, 15, 27, 28].

For problems of the form (1), [13] and [26] are perhaps the two most closely related papers to the present in the literature. In [13], a clever Lipschitzian optimization algorithm is presented which uses measurements with differing amounts of accuracy; this approach builds specifically upon accurate knowledge of the Lipchitz norm of the truth function. Proof of convergence of the algorithm proposed is provided in [13]. In [26], a promising Kriging-based algorithm of the surrogate management framework family is proposed, in a manner which increases the sampling of new measurements as convergence is approached. However, this method does not selectively refine existing measurements, which is a key contributor to the efficiency of the algorithm developed herein. Also, rigorous proof of convergence of the algorithm proposed in [26] is unavailable.

In this paper, a highly efficient and provably convergent new optimization approach is developed for problems of the form given in (1). The structure of the

remainder of the paper is as follows: Section 2 briefly reviews the key features of the Δ-DOGS(Z) algorithm developed in [5], upon which the present paper is built. Section 3 lays out all of the new elements that compose the new optimization approach, as well as the new algorithm itself, dubbed α-DOGS. Section 4 analyzes the convergence properties of the new algorithm, and establishes conditions which are sufficient to guaranty its convergence to the global minimum. Section 5 applies the new algorithm to a selection of model problems in order to illustrate its behavior. Some conclusions are presented in Section 6.

2. Delaunay-based optimization coordinated with a grid: Δ-DOGS(Z). This section presents a simplified version of the Δ-DOGS(Z) algorithm, the full version of which is given as Algorithm 2 of [5], where it is analyzed in detail. The Δ-DOGS(Z) algorithm is a grid-based acceleration of the Δ-DOGS algorithm originally developed in [6], and is designed to minimize problems in which accurate function evaluations are available, while avoiding an accumulation of unnecessary function evaluations on the boundary of the feasible domain.

The optimization problem we consider in this section is the minimization of an objective function $f(x)$, accurate evaluations of which are assumed to be available, in the feasible domain $L = \{x | x \in \mathbb{R}^n, a \leq x \leq b\}$. At each iteration of the simplified Δ-DOGS(Z) algorithm considered here, a metric based on an interpolation of existing function evaluations, and a model $e^k(x)$ of the "remoteness" of any point $x \in L$ from the available datapoints at that iteration (which in the Δ-DOGS(Z) algorithm characterizes the uncertainty of the interpolation) is minimized to obtain the location of the next point $x \in L$ at which the function will be evaluated. The interpolation and remoteness functions at iteration k are denoted here by $p^k(x)$ and $e^k(x)$, the latter of which is defined below. Note that the function $e^k(x)$ is called an "uncertainty" function in [4–6]; we use the name "remoteness" function for the same construction in the present paper, as this function plays a slightly different role in the sections that follow.

definition Consider S as a set of feasible points which includes the vertices of L, and Δ as a Delaunay triangulation of S. Then, for each simplex $\Delta_i \in \Delta$, the local remoteness function is defined as:

$$e_i(x) = R_i^2 - \|x - Z_i\|^2, \tag{2a}$$

where R_i and Z_i are the circumradius and circumcenter of Δ_i. The global remoteness function $e(x)$ is a piecewise quadratic function which is nonnegative everywhere and goes to zero at the datapoints, and is defined as follows:

$$e(x) = e_i(x) \quad \forall x \in \Delta_i. \tag{2b}$$

The remoteness function $e(x)$ has a number of properties which are established in Lemmas [2:5] in [6], as listed bellow.

Definition 1.
a. The remoteness function $e(x) \geq 0$ for all $x \in L$, and $e(x) = 0$ for all $x \in S$.
b. The remoteness function $e(x)$ is continuos and Lipschitz.
c. The remoteness function $e(x)$ is piecewise quadratic with Hessian of $-2\,I$.
d. The remoteness function $e(x)$ is equal to the maximum of the local remoteness functions:

$$e(x) = \max_{1 \leq i \leq n} e_i(x). \tag{3}$$

We now review the definition of the Cartesian grid over the feasible domain, as discussed further in [5].

definition Taking $N_\ell = 2^\ell$, the Cartesian grid of level ℓ over the feasible domain $L = \{x | a \le x \le b\}$, denoted L_ℓ, is defined as follows:

$$L_\ell = \left\{ x | x_i = a_i + \frac{b_i - a_i}{N_\ell} \cdot z, \quad z \in \{0, 1, \ldots, N_\ell\}, \quad i \in \{0, 1, \ldots, n\} \right\}.$$

The *quantization* of a point x onto the grid L_ℓ, denoted x_q^ℓ, is a point on the grid L_ℓ which has the minimum distance from x. Note that this quantization process might have multiple solutions; any of these solutions is acceptable. The maximum quantization error of the grid, δ_{L_ℓ}, is defined as follows:

$$\delta_{L_\ell} = \max_{x \in L_\ell} \|x - x_q^\ell\| = \frac{\|b - a\|}{2N_\ell}. \tag{4}$$

Three important properties of the Cartesian grid for the present purposes follow.

Definition 2.
a. The grid of level ℓ covering the feasible domain L in an n dimensional space has $(N_\ell + 1)^n$ grid points.
b. $\lim_{\ell \to \infty} \delta_{L_\ell} = 0$.
c. If x_q^ℓ is a quantization of x onto L_ℓ, then $A_a(x) \subseteq A_a(x_q^\ell)$, where $A_a(x)$ is the set of active constraints at x_0.

Algorithm 1 presents a strawman form of the Δ-DOGS(Z) algorithm. A significant refinement of this algorithm is presented as Algorithm 2 of [5], together with its proof of convergence and its implementation on model problems. The key refinement in [5] of the strawman algorithm presented here is the identification of two different sets of points in L, dubbed S_E (on which function evaluations are performed) and S_U (on which function evaluations are not yet performed); the latter set proves useful in [5] to regularize the triangulations. The algorithm proposed in §3 below effectively generalizes this notion, of evaluation points at which the function has been evaluated, and support points at which the function has not yet been evaluated, to the quantification and control over the *extent* of sampling performed for any given approximation of $f(x)$.

3. Delaunay-based optimization of a time-averaged value: α-DOGS. This section presents the essential elements of the new optimization algorithm, dubbed α-DOGS, which is designed to efficiently minimize a function $f(x)$ given by (1a) within the feasible domain L defined by (1b). We begin by introducing some fundamental concepts.

definitionTake S as a set of points x_i, for $i = 1, \ldots, M$, at which the function $f(x)$ in (1a) has been approximated; drawing a parallel to the nomenclature commonly used in estimation theory, we refer to any such approximation of $f(x)$, developed with a finite number of samples N_i, as a *measurement*, denoted y_i:

$$y_i = y(x_i, N_i) = \frac{1}{N_i} \sum_{k=1}^{N_i} g(x_i, k). \tag{5}$$

Any such measurement y_i has a finite uncertainty associated with it, which may made small by increasing N_i. For many problems, there is an initial transient such that, for $k < \bar{k}$, the assumption of stationarity of $g(x, t_k)$ is not valid. For such problems,

Algorithm 1 Strawman form of the grid-accelerated Delaunay-Based optimization algorithm, Δ-DOGS(Z), presented as Algorithm 2 in [5], which assumes that accurate function evaluations are available.

1: Set $k = 0$ and initialize ℓ. Take the set of initialization points S^0 as all vertices of the feasible domain L, together with any user-supplied points of interest (quantized onto the grid L_0), and perform function evaluations at all points in S^0.
2: Calculate (or, for $k > 0$, update) an appropriate interpolating function $p^k(x)$ through all points in S^k.
3: Calculate (or, for $k > 0$, update) a Delaunay triangulation Δ^k over all of the points in S^k.
4: Find z as a global minimizer of $s_c^k(x) = p^k(x) - K\, e^k(x)$ in L, and take z_ℓ as its quantization onto the grid L_ℓ.
5: If $z_\ell \notin S^k$, then take $S^{k+1} = S^k \cup \{z_\ell\}$ calculate $f(z_\ell)$, increment k, and repeat from 2.
6: Otherwise, take $S^{k+1} = S^k$, increment both k and ℓ, and repeat from 2.

the initial transient in the data can be detected using the approach developed in [1] and set aside, and the signal considered as stationary therafter. In such problems, to increase the speed of convergence of the statistics, the finite sum used for averaging the samples in (5) is modified to begin at $\bar{k}$ instead of beginning at 1. Since, for $k > \bar{k}$, $g(x, k)$ is statistically stationary, each measurement y_i is an *unbiased estimate* of the corresponding value of $f(x_i)$. We assume that a model for the *standard deviation* quantifying the uncertainty of this measurement, denoted $\sigma_i = \sigma(x_i, N_i)$, is also available. Since $g(x, t)$ is a stationary ergodic process, for any point $x_i \in L$,

$$\lim_{N_i \to \infty} y(x_i, N_i) = f(x_i), \quad \lim_{N_i \to \infty} \sigma(x_i, N_i) = 0. \tag{6}$$

If a stationary ergodic process $g(x, k)$ at some point $x_i \in L$ is *independent and identically distributed* (IID), then $\sigma(x_i, N_i) = \sigma(x_i, 1)/\sqrt{N_i}$; otherwise, estimates of $\sigma(x_i, N_i)$ can be developed using standard uncertainty quantification (UQ) procedures, such as those developed in [2, 3, 17, 22, 29]. The discrete-time process $g(x, k)$ may often be obtained by sampling a continuous-time process $g(x, t)$ at timesteps $t_k = kh$ for some appropriate sample interval h. For sufficiently large h, the samples of this continuous-time process $g(x, k)$ are often essentially IID; however, with the appropriate UQ procedures in place, significantly smaller sample intervals h will lead to a given degree of convergence in a substantially shorten period of time t, albeit with increased storage.definition Define S as a set of measurements y_i, for $i = 1, 2, \ldots, M$, at corresponding points x_i and with standard deviation σ_i. We will call a regression $p(x)$ for this set of measurements a *strict* regression if, for some constant β,

$$|p(x_i) - y_i| \leq \beta\, \sigma_i, \quad \forall\ 1 \leq i \leq M. \tag{7}$$

Based on the concepts defined above, Algorithm 2 presents our algorithm to efficiently and globally minimize a function of the form (1a) within a feasible domain defined by (1b). At each iteration k of this algorithm, S^k denotes the set of M points x_i, for $i = 1, 2, \ldots, M$, at which measurements have so far been made; for each point $x_i \in S^k$, $y_i = y(x_i, N_i)$ denotes the measured value, $\sigma_i = \sigma(x_i, N_i)$ denotes the

Algorithm 2 The new optimization algorithm, dubbed α-DOGS, for minimizing the function $f(x)$ in (1a) within the feasible domain L defined in (1b).

Definition 4.

1: Set $k = 0$ and initialize the algorithm parameters α, K, γ, β, ℓ, N^0, and N^δ as discussed in §3. Take the initial set of M sampled points, S^0, as the 2^n vertices of the feasible domain $L = \{x | a \le x \le b\}$ together with any user-supplied points of interest quantized onto the grid L_ℓ. Take $N_i = N^0$ for $i = 1, \ldots, M$, and compute an initial measurement $y_i = y(x_i, N^0)$ and corresponding uncertainty $\sigma_i = \sigma(x_i, N^0)$ for each point $x_i \in S^0$.

2: Calculate a strict regression $p^k(x)$ for all M available measurements.

3: Calculate (or, for $k > 0$, update) a Delaunay triangulation Δ^k over all of the points in S^k.

4: Determine x_j as the minimizer (and, j as the corresponding index), over all $x_i \in S^k$, of the discrete search function $s_d^k(x_i)$, defined as follows:

$$s_d^k(x_i) = \min\{p^k(x_i), 2\,y_i - p^k(x_i)\} - \alpha\,\sigma_i^k \quad \text{for } i = 1, \ldots, M. \tag{8}$$

5: Noting the definition of $e^k(x)$ in (2), determine z as the minimizer, over all $x \in L$, of the continuous search function $s_c^k(x)$, defined as follows:

$$s_c^k(x) = p^k(x) - K\,e^k(x) \quad \text{for } x \in L. \tag{9}$$

Denote z_ℓ as the quantization of z onto the grid L_ℓ.

6: If $s_c^k(z) > s_d^k(x_j)$ and $N_j < \gamma\,2^\ell$ where N_j is the current number of samples taken at x_j, then take N^δ additional samples at x_j, update $N_j \leftarrow N_j + N^\delta$, update the measurement $y_j = y(x_j, N_j)$ and uncertainty $\sigma_j = \sigma(x_j, N_j)$, increment k, and repeat from 2.

7: Otherwise, if $z_\ell \notin S^k$, then set $x_{M+1} = z_\ell$ and $S^{k+1} = S^k \cup \{x_{M+1}\}$, take $N_{M+1} = N^0$, compute the measurement $y_{M+1} = y(x_{M+1}, N^0)$ and uncertainty $\sigma_{M+1} = \sigma(x_{M+1}, N^0)$, increment M and k, and repeat from 2.

8: Otherwise, increment both ℓ and k, adjust the algorithm parameters such that $\alpha \leftarrow \alpha + \alpha^\delta$ and $K \leftarrow 2\,K$, and repeat from 2.

uncertainty of this estimate, and N_i quantifies the sampling performed thus far at point x_i. Note that the values of M, N_i, y_i, ℓ, α, and K are all updated from time to time as the iterations proceed, and are thus annotated with a k superscript at various points in the analysis of §4 to remove ambiguity. Akin to Algorithm 1, at iteration k, $p^k(x)$ is assumed to be a strict regression (for some value of β) of the current set of measurements y_i, and ℓ is the current grid level.

At each iteration of Algorithm 2, there are three possible situations, corresponding to three of the numbered iterations of this algorithm:

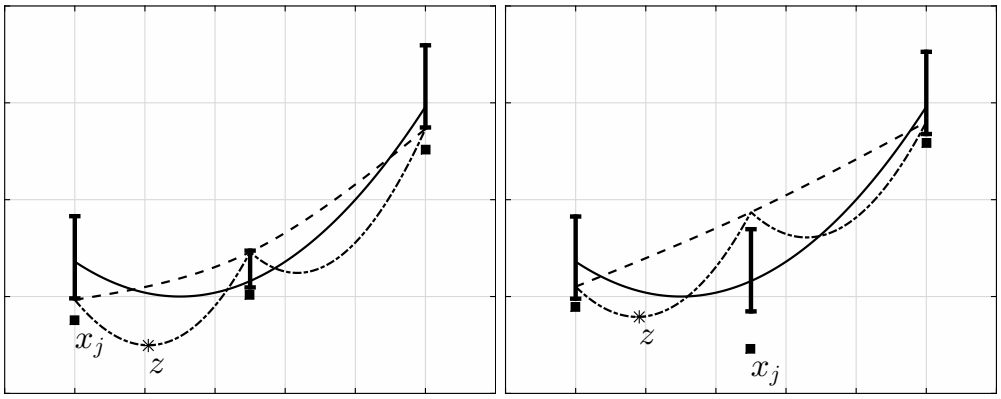
(6) The sampling of an existing measurement is increased. This is called a *supplemental sampling* iteration.

(7) A new point is identified, and an initial measurement at this point is added to the dataset. This is called an *identifying sampling* iteration.

(8) The mesh coordinating the problem is refined and the algorithm parameters α and K adjusted. This is called a *grid refinement* iteration.

Figure 1 illustrates supplemental sampling and identifying sampling iterations of Algorithm 2.

Algorithm 2 depends upon a handful of algorithm parameters, the selection of



(a) An identifying sampling iteration **(b)** A supplemental sampling iteration.

Figure 1: Representation of one iteration in Algorithm 2 in different situations: (solid line) truth function $f(x)$, (dashed line) the regression $p^k(x)$, (dash-dot line) continuous search function $s_c^k(x)$, (closed squares) $s_d^k(x)$, and (asterix) z. In figure (a), $s_c^k(z) < s_d^k(x_j)$; it is thus an identifying sampling iteration. In figure (b), $s_c^k(z) > s_d^k(x_j)$; it is thus an supplemental sampling iteration.

which affects its rate of convergence, explored in §5, though not its proof of convergence, established in §4. The remainder of this section discusses heuristic strategies to tune these algorithm parameters, noting that this tuning is an application-specific problem, and alternative strategies (based on experiment or intuition) might lead to more rapid convergence for certain problems.

The first task encountered during the setup of the optimization problem is the definition of the design parameters. Note that the feasible domain considered during the optimization process is characterized by simple upper and lower bounds for each design parameter; normalizing all design parameters to lie between 0 and 1 is often helpful.

The second challenge is to scale the function $f(x)$ itself, such that the range of the normalized function $f(x)$ over the feasible domain L is about unity. If an estimate of the actual range of $f(x)$ is not available a priori, we may estimate it at any given iteration using the available measurements. Following this approach, at any iteration k with available measurements $\{y_1, y_2, \ldots, y_M\}$, all measurements y_i, as well as the corresponding uncertainty of these measurements σ_i, may be scaled by a factor r_s wherever used in iteration k of Algorithm 2 where, for that iteration, r_s is computed such that

$$r = \frac{1}{\max_{1\le i\le M}\{y_i\} - \min_{1\le i\le M}\{y_i\}}, \quad r_s = r_l + R(r - r_l) - R(r - r_u),$$

where $R(x)$ is the ramp function. So defined, r_s is a "saturated" version of the factor r, constrained to lie in the range $r_l \le r_s \le r_u$. Note that scaling the y_i and σ_i does not interfere with the proof of the convergence of the algorithm, provided in §4, but can improve its performance. In the numerical simulations performed in §5, we take $r_l = 10^{-3}$ and $r_u = 10^3$.

For problems which are IID with no initial transient, $N^0 = N^\delta = 1$ is a reasonable starting point; increasing N^0 and N^δ ultimately reduces the number of iterations of the algorithm (and, thus, the number of Delaunay triangulations) required for convergence, but generally increases slightly the total amount of sampling performed. Suggested values of other algorithm parameters, which work well in the numerical simulations reported in §5 but the values of which do not affect the proof of convergence provided in §4, include $\alpha^0 = \alpha^\delta = 0.5$, $K^0 = 0.5$, $\ell^0 = 3$, $\beta = 4$, and $\gamma = 100$.

4. Analysis of α-DOGS. We now analyze the convergence of Algorithm 2. We first present some preliminary definitions.

definition The point $\eta^k \in S$ is called the *candidate* point at iteration k if

$$\eta^k = \text{argmin}_{z \in S^k}\{y_k(z) + \alpha^k \sigma_k(z)\}. \tag{10}$$

Define $f(x^*)$ as the global minimum of $f(x)$ in L, then the *regret* is defined as

$$r_k = f(v_k) - f(x^*). \tag{11}$$

The definition of the regret given above is common in the optimization literature (see, e.g., [8, 13, 27]). We show in this section that, under the following assumptions, the regret of the optimization process governed by Algorithm 2 will converge to zero:

A constant $\hat{K}$ exist such that, for all $k > 0$ and $x \in L$,

$$-2\,\hat{K} \le \nabla^2 p^k(x) \le 2\,\hat{K}, \quad -2\,\hat{K} \le \nabla^2 f(x) \le 2\,\hat{K}.$$

Assumption 2. *There is a real positive function $E(x) : \mathbb{R}^+ \to \mathbb{R}^+$, which has the following properties:*

a. $E(x)$ is continuous and monotonically increasing.

b. $E(x)$ is bounded such that

$$E(x) \le Q, \quad \sup_{x \in \mathbb{R}^+} E(x) = Q. \tag{12}$$

c. For all $x \in L$ and $N \in \mathbb{N}$, we have:

$$\left|\frac{1}{N}\sum_{k=1}^{N} g(x,k) - f(x)\right| = |y(x,N) - f(x)| \le E(\sigma(x,N)). \tag{13}$$

d. $\lim_{x \to 0^+} E(x) \to 0$.

Assumption 3. *There are real numbers $\alpha > 0$ and $\theta \in (0\,1]$, such that*

$$\sigma(x,N) \le \alpha N^{-\theta} \quad \forall x \in \mathbb{R},\ N \in \mathbb{N}. \tag{14}$$

Note that, if the stationary process $g(x,k)$ has a short-range dependence, like ARMA processes, the parameter $\theta = 0.5$. However, for different θ, this general model can also handle stationary processes with long-range dependence, like Fractional ARMA processes. Moreover, at any given point $x \in L$, $\sigma(x,N)$ is a monotonically nonincreasing function of N. This condition means that, by increasing the averaging interval, the uncertainty of the estimate must not increase.

Assumption 2 is a stronger condition than ergodicity of $g(x,k)$. Recall that ergodicity of $g(x,k)$ is equivalent to the convergence of $y(x,N)$ to $f(x)$ as $N \to \infty$, which is the straightforward outcome of Assumptions 2 and 3. In Assumption 2, the

convergence of the sample mean is assumed to be bounded by a function of $\sigma(x, N)$ of the form specified.

lemma For any point $x \in L$ and real positive number $0 < \varepsilon < Q$,

$$|y(x, N) - f(x)| - \frac{Q}{E^{-1}(\varepsilon)}\,\sigma(x, N) \leq \varepsilon, \tag{15}$$

Proof. If $\sigma(x, N) \leq E^{-1}(\varepsilon)$ then, since $E(x)$ is an increasing function, by (13), we have:

$$E(\sigma(x, N)) \leq E(E^{-1}(\varepsilon)) = \varepsilon \quad \Rightarrow \quad |y(x, N) - f(x)| \leq \varepsilon.$$

Otherwise, $\sigma(x, N) > E^{-1}(\varepsilon)$; thus, again by (13), we have

$$\frac{Q}{E^{-1}(\varepsilon)}\,\sigma(x, N) \geq Q \quad \Rightarrow \quad |y(x, N) - f(x)| - Q \leq E(\sigma_N) - Q \leq 0,$$

Thus, (15) is verified for both cases.

lemma During the execution of Algorithm 2, there are an infinite number of mesh refinement iterations.

Proof. This lemma is shown by contradiction. If Algorithm 2 has a finite number of mesh refinement iterations, then there is an integer number $\bar{\ell}$ such that the mesh $L_{\bar{\ell}}$ contains all datapoints obtained by the algorithm. Since the number of datapoints on this mesh is finite, only a finite number of points must be considered, which leads to having a finite number of identifying sampling iterations.

Since the number of identifying sampling and mesh refinement iterations are finite, there must be an infinite number of supplemental sampling iterations. At each supplemental sampling iteration, the averaging length of the estimate at an existing datapoint is incremented by $N^{\delta} \geq 1$. Since only a finite number of points is considered, a datapoint exists for which the estimate is improved for an infinite number of supplemental sampling iterations. As a result, there is an supplemental sampling iteration, such that $N_j > \gamma 2^{\bar{\ell}}$, which is in contradiction with the assumption of having finite number of mesh refinement iterations.

Note that the following short Lemma and proof, which are necessary for this development, are copied directly from [5].

lemma Consider $G(x)$ as a twice differentiable function such that $\nabla^2 G(x) - 2\,K_1 I \leq 0$, and $x^* \in L$ as a local minimizer of $G(x)$ in L. Then, for each $x \in L$ such that $A_a(x^*) \subseteq A_a(x)$, we have:

$$G(x) - G(x^*) \leq K_1 \|x - x^*\|^2. \tag{16}$$

Proof. Define function $G_1(x) = G(x) - K_1 \|x - x^*\|^2$. By construction, $G_1(x)$ is concave; therefore,

$$\begin{aligned} G_1(x) &\leq G_1(x^*) + \nabla G_1(x^*)^T (x - x^*), \\ G_1(x^*) &= G(x^*), \quad \nabla G_1(x^*) = \nabla G(x^*), \\ G(x) &\leq G(x^*) + \nabla G(x^*)^T (x - x^*) + K_1 \|x - x^*\|^2. \end{aligned}$$

Since the feasible domain is a bounded domain, the constrained qualification holds; therefore, x^* is a KKT point. Therefore, using $A_a(x^*) \subseteq A_a(x)$ leads to $\nabla G(x^*)^T (x - x^*) = 0$, which verifies (16).

lemma Consider z, x_j, and x^* as global minimizers of $s_c^k(x)$, $s_d^k(x)$, and $f(x)$, respectively. Note that $s_d^k(x)$ is only defined for the points in S^k, but $s_c^k(x)$ and $f(x)$ are defined over the feasible domain L. Define M_k as:

$$M_k = \min\{s_c^k(z) - f(x^*), s_d^k(x_j) - f(x^*)\}. \tag{17}$$

Then,

$$\limsup_{k\to\infty} M_k \leq 0. \tag{18}$$

Proof. By Lemma 7, there are infinite number of mesh refinement iterations during the execution of Algorithm 2. Thus,

$$\lim_{k\to\infty} K^k = \infty, \quad \lim_{k\to\infty} \alpha^k = \infty. \tag{19}$$

As a result, for any $0 < \varepsilon < Q$, there is a k_ε such that, if $k > k_\varepsilon$, then

$$K^k \geq 3\,\hat{K} \quad \text{and} \quad \alpha^k \geq \frac{2\,Q}{E^{-1}(\varepsilon)}. \tag{20}$$

Consider $\Delta_{x^*}^k$ as a simplex in Δ^k, a Delaunay triangulation for S^k, that contains x^*. Define $M(x) : \Delta_{x^*}^k \to \mathbb{R}$ as the unique linear function in $\Delta_{x^*}^k$ such that

$$M(V_j^k) = 2\,f(V_j^k) - p^k(V_j^k),$$

where V_j^k are the vertices of $\Delta_{x^*}^k$. Define $G(x) : \Delta_{x^*}^k \to \mathbb{R}$ as follows:

$$G(x) = s_c^k(x) + M(x) - 2\,f(x) = p^k(x) + M(x) - 2\,f(x) - K^k\,e^k(x).$$

By construction, $G(V_j^k) = 0$. Moreover:

$$\nabla^2 G(x) = \nabla^2\{p^k(x) - 2\,f(x)\} + 2\,K^k\,I.$$

Using Assumption 1 and (20), $G(x)$ is strictly convex in simplex $\Delta_{x^*}^k$. Since $G(x) = 0$ at the vertices of $\Delta_{x^*}^k$, then $G(x^*) \leq 0$.

Moreover, since $M(x)$ is a linear function, then

$$\min_{x\in S^k}[2\,f(x) - p^k(x)] \leq \min_{1\leq j\leq n+1}[2\,f(V_j^k) - p^k(V_j^k)] \leq M(x^*),$$

$$s_d^k(x) \leq [2\,f(x) - p^k(x)] + 2\,(y_k(x) - f(x)) - \alpha^k\sigma(x). \tag{21}$$

Using (15) in Lemma 6 and (20) leads to:

$$2\,y_k(x) - 2\,f(x) - \alpha^k\sigma(x) \leq 2\,\varepsilon. \tag{22}$$

Combining (21) and (22) leads to:

$$s_d^k(x) \leq [2\,f(x) - p^k(x)] + 2\,\varepsilon.$$

Since x_j is the minimizer of the $s_d^k(x)$,

$$s_d^k(x_j) \leq \min_{x\in S^k}[2\,f(x) - p^k(x)] + 2\,\varepsilon \leq M(x^*) + 2\,\varepsilon.$$

Furthermore, z is the global minimizer of $s_c^k(x)$ and $G(x^*) \leq 0$; therefore,

$$s_c^k(z) \leq s_c^k(x^*) \leq 2\, f(x^*) - M(x^*),$$
$$s_c^k(z) + s_d^k(x_j) \leq 2\, f(x^*) + 2\, \varepsilon. \tag{23}$$

Thus, for any $\varepsilon > 0$ and $k > \hat{k}_\varepsilon$, (23) is satisfied; therefore, (18) is verified.

lemma If $\{k_1, k_2, \dots\}$ are the mesh refinement iterations of Algorithm 2, then

$$\limsup_{i\to\infty} \left\{ y(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) - f(x^*) + \alpha^{k_i} \sigma(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) \right\} \leq 0, \quad \textit{and} \tag{24a}$$

$$\lim_{i\to\infty} \sigma(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) = 0, \tag{24b}$$

where η^{k_i} is the candidate point at iteration k_i and x^* is a global minimizer of $f(x)$ in L.

Proof. Consider z as a global minimizer of $s_c^{k_i}(x)$ in L, and z_ℓ as its quantization on L_ℓ. Since iteration k_i is a mesh refinement, $z_\ell \in S^{k_i}$. Consider $\Delta_j^{k_i}$ as a simplex in the Delaunay triangulation Δ^{k_i} which contains z. By property (d) of the remoteness function $e^{k_i}(x)$,

$$e^{k_i}(z_\ell) \geq e_j^{k_i}(z_\ell), \quad e^{k_i}(z) = e_j^{k_i}(z),$$
$$s_c^{k_i}(z_\ell) - s_c^{k_i}(z) = p^{k_i}(z_\ell) - p^{k_i}(z) + K^{k_i}(e^{k_i}(z) - e^{k_i}(z_\ell)),$$
$$s_c^{k_i}(z_\ell) - s_c^{k_i}(z) \leq p^{k_i}(z_\ell) - p^{k_i}(z) + K^{k_i}(e_j^{k_i}(z) - e_j^{k_i}(z_\ell)).$$

By Property (c) of Remark 2 concerning the Cartesian grid quantizer, $A_a(z) \subseteq A_a(z_\ell)$. According to Assumption 1 and Property (c) of the remoteness function introduced in Definition 1, $\nabla^2\{p^{k_i}(x) - K^{k_i} e_j^{k_i}(x)\} - \{\hat{K} + 2\, K^{k_i}\} I \leq 0$; thus, by Lemma 8 and the fact (see Lemma 5 in [6] for proof) that z globally minimizes $p^{k_i}(x) - K^{k_i} e_j^{k_i}(x)$,

$$s_c^{k_i}(z_\ell) - s_c^{k_i}(z) \leq \{\hat{K} + 2\, K^{k_i}\} \|z_\ell - z\|^2. \tag{25}$$

Define δ_{k_i} as the maximum quantization error at iteration k_i, then $\|z_\ell - z\| \leq \delta_{k_i}$. On the other hand, $z_\ell \in S^{k_i}$, which leads to $s_c^{k_i}(z_\ell) = p^{k_i}(z_\ell)$, and

$$p^{k_i}(z_\ell) \leq s_c^{k_i}(z) + \{\hat{K} + 2\, K^{k_i}\} \delta_{k_i}^2. \tag{26}$$

At each mesh refinement iteration of Algorithm 2, there are two possibilities. In the first case, $s_c^{k_i}(z) \leq s_d^{k_i}(x_j)$; since x_j is a minimizer of $s_d^{k_i}(x)$, then

$$p^{k_i}(z_\ell) \leq s_d^{k_i}(z_\ell) + \{\hat{K} + 2\, K^{k_i}\} \delta_{k_i}^2,$$
$$s_d^{k_i}(z_\ell) \leq p^{k_i}(z_\ell) - \alpha^{k_i} \sigma(z_\ell, N_{z_\ell}^{k_i}),$$
$$\sigma(z_\ell, N_{z_\ell}^{k_i}) \leq \frac{\{\hat{K} + 2\, K^{k_i}\}}{\alpha^{k_i}} \delta_{k_i}^2. \tag{27}$$

Using (17) (see Lemma 9) and (26) leads to

$$p^{k_i}(z_\ell) - f(x^*) \leq M_{k_i} + \{\hat{K} + 2\, K^{k_i}\} \delta_{k_i}^2. \tag{28}$$

Since the regression is strict,

$$y(z_\ell, N_{z_\ell}^{k_i}) - p^{k_i}(z_\ell) \leq \beta\, \sigma(z_\ell, N_{z_\ell}^{k_i}). \tag{29}$$

Using (27), (28), and (29) leads to

$$y(z_\ell, N_{z_\ell}^{k_i}) - f(x^*) \leq M_{k_i} + \{\hat{K} + 2\,K^{k_i}\}\delta_{k_i}^2 + \beta\Big[\frac{\{\hat{K} + 2\,K^{k_i}\}}{\alpha^{k_i}}\delta_{k_i}^2\Big]. \tag{30}$$

In the second case, $s_c^{k_i}(z) > s_d^{k_i}(x_j)$, then by the construction of M_{k_i} (see (17)),

$$s_d^{k_i}(x_j) - f(x^*) = M_{k_i}. \tag{31}$$

Moreover, since iteration k_i is mesh refinement, then the sampling $N_j \geq \gamma 2^\ell$. Thus, using Assumption 3,

$$\sigma(x_j, N_{x_j}^{k_i}) \leq \alpha\,\gamma^{-\theta} 2^{-\theta\ell}. \tag{32}$$

Furthermore, the regression $p^{k_i}(x)$ is strict which leads to:

$$y(x_j, N_{x_j}^{k_i}) - s_d^{k_i}(x_j) \leq (\beta + \alpha^{k_i})\,\sigma(x_j, N_{x_j}^{k_i}) \tag{33}$$

Using (31), (32), and (33) leads to

$$y(x_j, N_{x_j}^{k_i}) - f(x^*) \leq M_{k_i} + (\beta + \alpha^{k_i})\alpha\gamma^{-\theta} 2^{-\theta\ell}. \tag{34}$$

Note that η^{k_i} is the candidate point at iteration k_i. Thus, using (27), (30), (32), and (34), and the construction of candidate point (see Definition 5),

$$\begin{aligned} y(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) - f(x^*) + \alpha^{k_i}\sigma(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) \leq M_{k_i} + \\ \max\Big\{(\beta + \alpha^{k_i})\alpha\gamma^{-\theta}2^{-\theta\ell}, (\hat{K} + 2\,K^{k_i})\delta_{k_i}^2 + \beta\Big[\frac{(\hat{K} + 2\,K^{k_i})}{\alpha^{k_i}}\delta_{k_i}^2\Big]\Big\} \\ +\alpha^{k_i}\max\Big\{\alpha\gamma^{-\theta}2^{-\theta\ell}, \frac{(\hat{K} + 2\,K^{k_i})}{\alpha^{k_i}}\delta_{k_i}^2\Big\}. \end{aligned} \tag{35}$$

On the other hand,

$$\delta_{k_i} = \frac{\|b - a\|}{2^{\ell_0 + i}}, \quad \alpha^{k_i} = \alpha^0 + i\,\alpha^\delta, \quad K^{k_i} = K_0\,2^i, \quad \ell^{k_i} = \ell^0 + i. \tag{36}$$

By substituting (36) in (35) and using (18) (see Lemma 9), (24a) is verified. Furthermore, using Assumption 2, we have

$$|y(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) - f(\eta^{k_i})| \leq E(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) \leq Q. \tag{37}$$

Thus, using (24a), $f(\eta^{k_i}) - f(x^*) > 0$, and (37) leads to

$$\limsup_{i\to\infty}\Big\{ - Q + \alpha^{k_i}\sigma(\eta^{k_i}, N_{\eta^{k_i}}^{k_i})\Big\} \leq 0.$$

Since $\sigma(\eta^{k_i}, N_{\eta^{k_i}}^{k_i}) \geq 0$ and $\lim_{i\to\infty}\alpha^{k_i} = \infty$, (24b) is verified.

theoremConsider η^k as the candidate point at iteration k of Algorithm 2, then

$$\lim_{k\to\infty} f(\eta^k) = f(x^*). \tag{38}$$

Theorem 11. *Proof.* At any iteration $k > k_1$, take $k_i < k$ as the most recent mesh refinement iteration of Algorithm 2. Then $\eta^{k_i} \in S^k$, and

$$y(\eta^k, N_{\eta^k}^k) + \alpha^k\sigma(\eta^k, N_{\eta^k}^k) \leq y(\eta^{k_i}, N_{\eta^{k_i}}^k) + \alpha^k\sigma(\eta^{k_i}, N_{\eta^{k_i}}^k). \tag{39}$$

Using Assumption 2 leads to:

$$|y(\eta^{k_i}, N^k_{\eta^{k_i}}) - y(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})| \le E(\sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})) + E(\sigma(\eta^{k_i}, N^k_{\eta^{k_i}})),$$
$$y(\eta^k, N^k_{\eta^k}) + \alpha^k \sigma(\eta^k, N^k_{\eta^k}) \le y(\eta^{k_i}, N^{k_i}_{\eta^{k_i}}) + \alpha^k \sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}}) +$$
$$E(\sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})) + E(\sigma(\eta^{k_i}, N^k_{\eta^{k_i}})).$$

By construction, since the sampling at η^{k_i} at iteration k is greater than or equal to its sampling at iteration k_i, $\sigma(\eta^{k_i}, N^k_{\eta^{k_i}}) \le \sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})$. Since the function $E(x)$ is nondecreasing,

$$y(\eta^k, N^k_{\eta^k}) + \alpha^k \sigma(\eta^k, N^k_{\eta^k}) \le y(\eta^{k_i}, N^{k_i}_{\eta^{k_i}}) + \alpha^k \sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}}) + 2\, E(\sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})).$$

Using (24) in Lemma 10, Assumption 2, and $\alpha^k = \alpha^{k_i} + \alpha^\delta$, leads to:

$$\limsup_{k\to\infty} y(\eta^k, N^k_{\eta^k}) - f(x^*) + \alpha^k \sigma(\eta^k, N^k_{\eta^k}) \le \limsup_{k\to\infty} \left\{\alpha^\delta \sigma(\eta^{k_i}, N^{k_i}_{\eta^{k_i}})\right\} = 0. \tag{40}$$

Similar to the proof of (24b), it is thus again easy to show

$$\lim_{i\to\infty} \sigma(\eta^k, N^k_{\eta^k}) = 0.$$

On the other hand, based on Assumption 2, and optimality of $f(x^*)$

$$f(\eta^k) + \alpha^k \sigma(\eta^k, N^k_{\eta^k}) - f(x^*) - E(\sigma(\eta^k, N^k_{\eta^k})) \le y(\eta^k, N^k_{\eta^k}) + \alpha^k \sigma(\eta^k, N^k_{\eta^k}) - f(x^*),$$
$$\lim_{k\to\infty} E(\sigma(\eta^k, N^k_{\eta^k})) = 0,$$
$$\limsup_{k\to\infty} f(\eta^k) - f(x^*) \le 0.$$

Since $f(\eta^k) - f(x^*) \ge 0$, (38) is verified.

5. Results. We now illustrate the performance of Algorithm 2 on some representative examples. The function $g(x,k)$ in (1a) is assumed to be a discrete-time statistically stationary random ergotic process. In this section, we further assume that $g(x,k)$ is IID in the index k, and that the variation of $g(x,k)$ from the truth function $f(x)$ is homogeneous in x. In particular, we take $\sigma(x_i, 1) = 0.3$, and

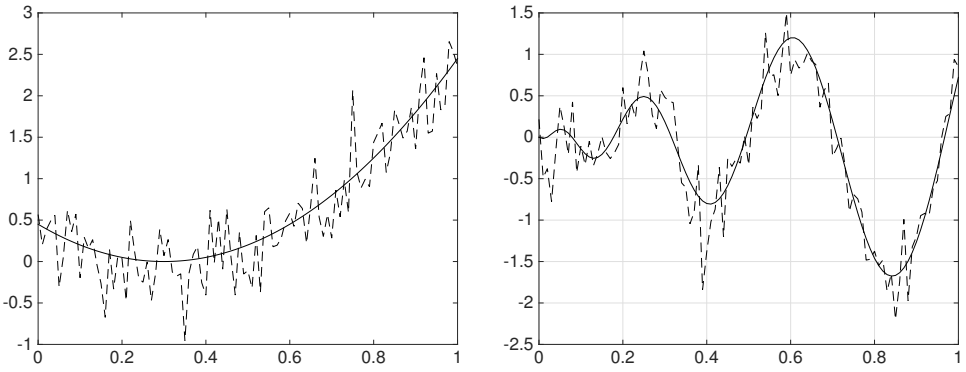
$$g(x,k) = f(x) + v_k \quad \text{where } v_k = \mathcal{N}(0, 0.09).$$

In this section, two different test functions for $f(x)$ are considered within the simple feasible domain $L = \{x | 0 \le x_i \le 1\ \forall i\}$, the *shifted parabolic function*

$$f(x) = \sum_{i=1}^{n} (x_i - 0.3)^2, \tag{41}$$

with a global minimizer in L of $x_i^* = 0.3$ and a corresponding global minimum of $f(x^*) = 0$, and the *scaled Schwefel fuction*

$$f(x) = 1.6759\, n - \sum_{i=1}^{n} \frac{x_i}{2} \sin(500\, |x_i|), \tag{42}$$



(a) Shifted parabolic function (41). **(b)** Scaled Schwefel fuction (42).

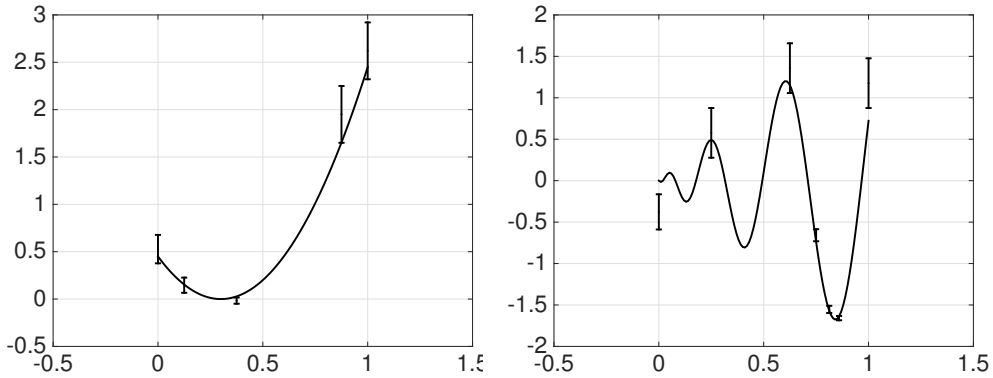
Figure 2: Illustration of test problems (41) and (42). (solid line) truth function $f(x)$, and (dashed line) a set of measurements y_i computed with a single sample at each measurement, $N_i = 1$.

with a global minimizer in L of $x_i^* = 0.8419$ and a corresponding global minimum of $f(x^*) = -1.6759\,n$. We will consider these two functions in $n = 1$, 2, and 3 dimensions.

One-dimensional representations of these functions are illustrated in Fig 2: for the shifted parabolic function (41), the truth function (unknown to the optimization algorithm) is a simple parabola, whereas for the scaled Schwefel fuction (42), the truth function is a smooth nonconvex function with four local minima. Note that the perturbations present in several measurements of these functions, computed with finite N_i, result in a complicated, nonsmooth, nonconvex behavior. This paper shows how to efficiently minimize such functions based only on such noisy measurements, automatically refining the measurements (by increasing the sampling) as convergence is approached.

The optimizations are initialized with measurements of sample length $N^0 = 1$ at the vertices of L. Figure 3 illustrates the application of Algorithm 2 after $k = 100$ iterations in the 1D case, taking $N^0 = N^\delta = 1$ additional sample (at either a new measurement point, or at an existing measurement point) at each iteration of the algorithm. In Figure 3a, the sampling N_i after $k = 100$ iterations (plus the 2 initial sample points, for a total of 102 samples) at the $M = 5$ measured points y_i indicated, enumerated from left to right, is $\{4, 14, 82, 1, 1\}$; in Figure 3b, the sampling N_i after 100 iterations at the 7 measured points indicated is $\{2, 1, 1, 9, 23, 65, 1\}$. Both results clearly show that the algorithm focuses the bulk of its sampling in the immediate vicinity of the minimum, where the accuracy of the measurements is especially important, while avoiding unnecessary sampling far from the minimum, where the accuracy of the measurements is of reduced importance. It is also seen that more exploration is performed for the scaled Schwefel function than for the shifted parabolic function, as a result of its more complex underlying trend.

Since the function evaluation process in these tests has a stochastic component, Algorithm 2 was next applied an ensemble of three separate tests, for both model problems discussed above, in each of three different cases with increasingly higher



(a) Shifted parabolic function (41). **(b)** Scaled Schwefel function (42).

Figure 3: Illustration of Algorithm 2 on model problems (41) and (42) in 1D after 100 iterations, taking $N^0 = N^\delta = 1$: (solid line) the truth function $f(x)$, and (error bars) the 66 percent confidence intervals of the measurements.

dimension (that is, $n = 1$, $n = 2$, and $n = 3$). The convergence histories of these simulations are illustrated in Figures 4 and 5.

To better quantify the performance of the algorithm proposed, we now introduce the following concept.

definitionAssume that the stationary process $g(x, k)$ is IID, and that the nominal variance $\sigma(x_i, 1) = \sigma_0$ for all points $x_i \in L$. As mentioned in Remark 3, denoting N_i as the total number of samples taken at point x_i, the uncertainty of the corresponding measurement y_i given by (5) is $\sigma_i = \sigma(x_i, N_i) = \sigma_0/\sqrt{N_i}$. If we assume that all of sampling of the algorithm is performed at a single point, the uncertainty of this single measurement after k samples would thus be $\sigma_0/\sqrt{k}$, which we refer to as the *reference error*. This function is indicated in Figures 4 and 5 by a solid line of slope $-1/2$ in log-log coordinates. It is remarkable to note that, in all 18 of the optimizations reported in Figures 4 and 5, in which we have again taken 1 new sample at each iteration, the regret function of Algorithm 2 is eventually diminished to a value substantially smaller than the reference error. That is, the value of the regret at the end of these optimizations is actually substantially *less* than the uncertainty of a single measurement, assuming that all of the sampling is done at a single point. Figures 4 and 5 also report the number of datapoints which are considered by the optimization algorithm as the iterations proceed. This number is important in optimization problems for which the function evaluations are obtained from simulations which have an (expensive) initial transient, which must be set aside before sampling the statistic of interest, as discussed further in Remark 3. It is observed, as in the 1D case illustrated in Figure 3, that the number of datapoints that are considered for the shifted parabolic function is less than that for the scaled Schwefel function. Further, the regret function converges faster to the general proximity of the global solution, in about 10 to 50 iterations, for the shifted parabolic function. This result is reasonable, since the underlying function in the shifted parabolic

case is simpler. This paper presents a new optimization algorithm, dubbed α-DOGS, for the minimization of functions given by the infinite-time average of stationary ergodic processes in the computational or experimental setting. Two search functions are considered at each iteration. The first is a continuous search functions, $s_c^k(x)$, defined over the entire feasible space $x \in L$, combining a strict regression $p^k(x)$ of the available datapoints together with a remoteness function characterizing the distance of any given point in the feasible domain from the nearest measurements, and built on the framework of a Delaunay triangulation over all available measurements at that iteration. The second is a discrete search function, $s_d^k(x_i)$, defined over the available measurements $x_i \in S^k$. A comparison between the minima of these two search functions is made in order to decide between further sampling (and, therefore, refining) an existing measurement, or sampling at a new point in parameter space. The method developed builds closely on the Delaunay-based Derivative-free Optimization via Global Surrogates algorithm, dubbed Δ-DOGS, proposed in [4–6]. The convergence of the algorithm is established in problems for which

Definition 12.

a. The underlying truth (infinite-time averaged) function, as well as the regressions computed at each iteration k, are twice differentiable.

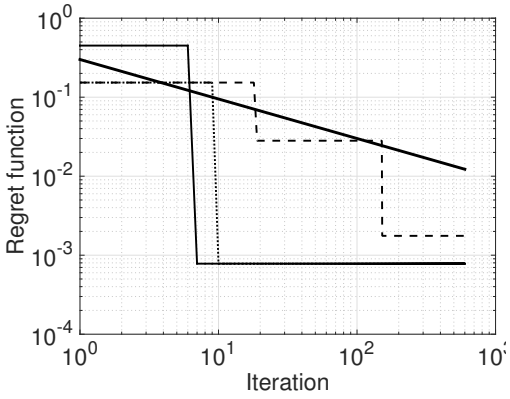
b. The stationary process $g(x, k)$ upon which the truth function $f(x)$ is generated, in (1a), is ergodic, and the convergence of the averaging process to the underlying truth function is bounded by a monotonic function of a computable uncertainty function (see Assumption 2).

c. The uncertainty of the time averaging process decays exponentially to zero (see Assumption 3); this is true for almost all stationary models of random processes.

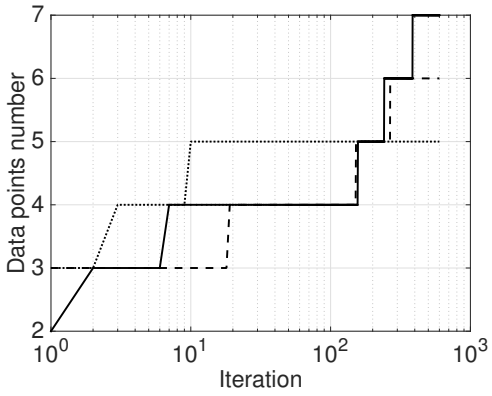
6. Conclusions. The α-DOGS algorithm performs and refines measurements with different amounts of sampling in different locations in the feasible region of parameter space as necessary. By so doing, the total cost of the optimization process is substantially reduced as compared with using existing derivative-free optimization strategies, with the same amount of sampling at different locations in parameter space. Computational experiments demonstrate that the algorithm developed ultimately devotes most of its sampling time to points in parameter space near to the global minimum. Further, these computational experiments indicate that the regret function (see Definition 5) eventually diminishes to a value that is actually substantially less than the uncertainty of a single measurement, assuming that all of the sampling is done at a single point.

In future work, the α-DOGS algorithm will be applied to additional benchmark and application-based optimization problems, including shape optimization for airfoils and hydrofoils. For problems in which the function is determined computationally (from, e.g., numerical simulations of turbulent flows), the extension of the present framework to, as convergence is approached, simultaneously (a) refine the computational grid, and (b) increase the measurement sampling, is also under development.

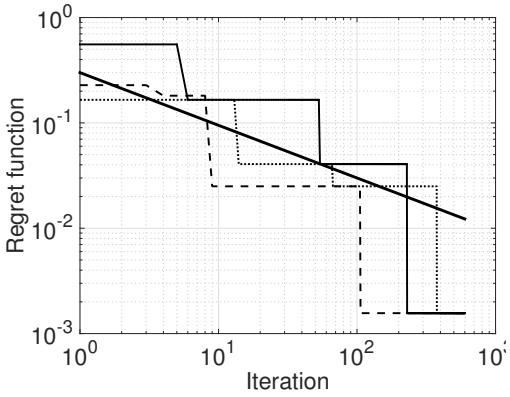
Appendix: Polyharmonic spline regression. The algorithm described in this paper depends upon a smooth regression $p^k(x)$ (see Assumption 1). The best technique for computing the regression is problem dependent. As with [4–6], a key advantage of our Delaunay-based approach in the present work is that it facilitates the use of *any* suitable regression technique, subject to it satisfying the "strict" regression property given in Definition 4. Since our numerical tests all implement the



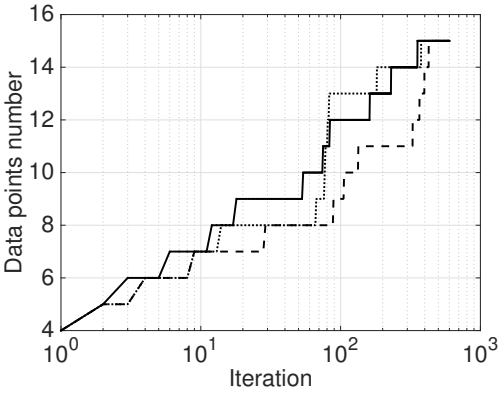
(a) The regret function in 1D.



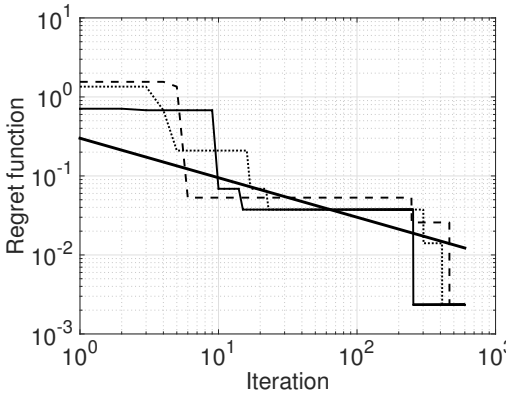
(b) Total number of datapoints in 1D.



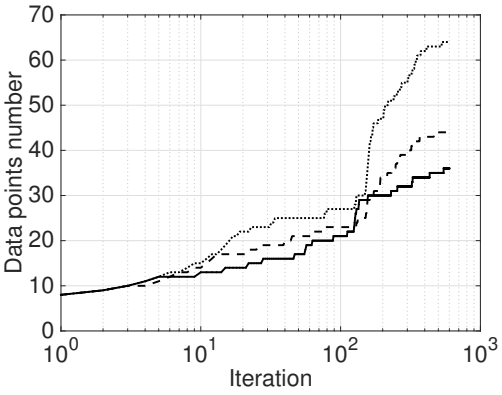
(c) The regret function in 2D.



(d) Total number of datapoints in 2D.

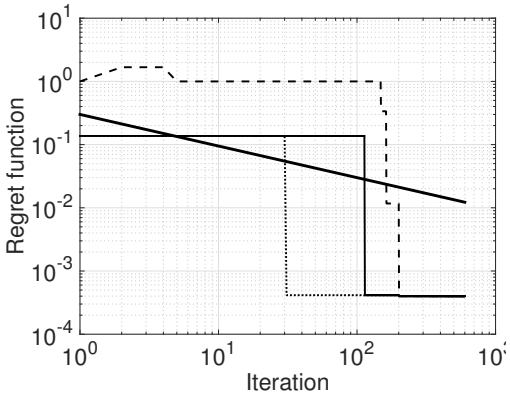


(e) The regret function in 3D.

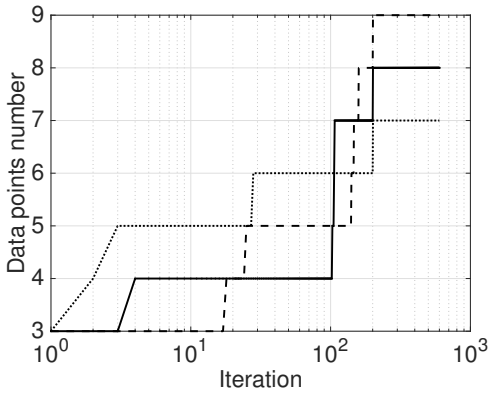


(f) Total number of datapoints in 3D.

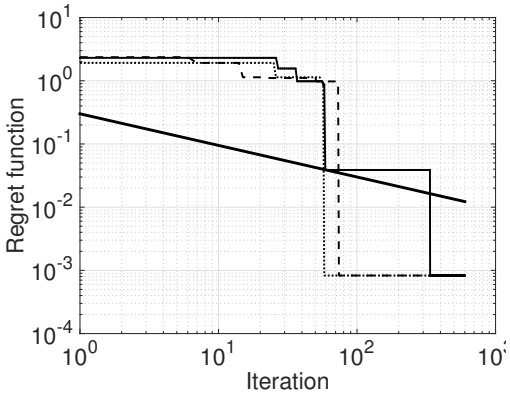
Figure 4: Implementation of Algorithm 2 on parabolic test problem (41).



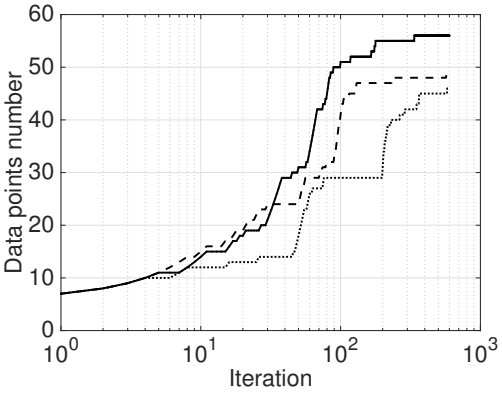
(a) The regret function in 1D.



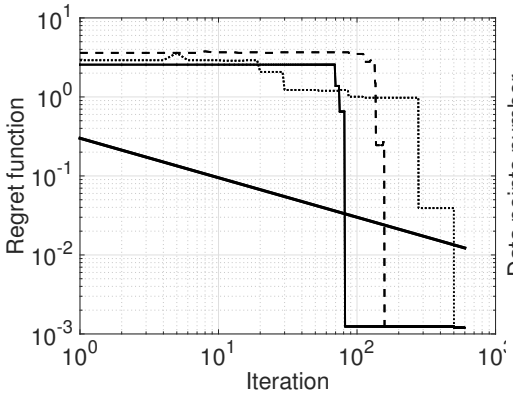
(b) Total number of datapoints in 1D.



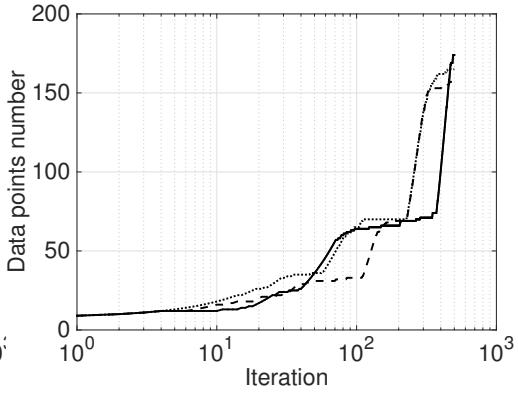
(c) The regret function in 2D.



(d) Total number of datapoints in 2D.



(e) The regret function in 3D.



(f) Total number of datapoints in 3D.

Figure 5: Implementation of Algorithm 2 on Schwefel test problem (42).

polyharmonic spline regression technique, the derivation of this regression technique is briefly explained in this appendix; additional details may be found in [30].

The polyharmonic spline regression $p(x)$ of a function $f(x)$ in $\mathbb{R}^n$ is defined as a weighted sum of a set of radial basis functions $\varphi(r)$ built around the location of each measurement point, plus a linear function of x:

$$p(x) = \sum_{i=1}^{N} w_i \, \varphi(r) + v^T \begin{bmatrix} 1 \\ x \end{bmatrix}, \tag{43}$$
$$\text{where} \quad \varphi(r) = r^3 \quad \text{and} \quad r = \|x - x_i\|.$$

The weights w_i and v_i represent N and $n+1$ unknowns. Assume that $\{y(x_1), y(x_2), \ldots, y(x_n)\}$ is the set of measurements, with standard deviations $\{\sigma_1, \sigma_2, \ldots, \sigma_2\}$. The w_i and v_i coefficients are computed by minimizing the following objective function, which expresses is a tradeoff between the fit to the observed data and the smoothness of the regressor:

$$L_p(x) = \sum_{i=1}^{N} \Big[\frac{(p(x_i) - y(x_i))}{\sigma_i} \Big]^2 + \lambda \int_B |\nabla^m p(x)|, \tag{44}$$

where B is a large box domain containing all of the x_i, and $\nabla^m p(x)$ is the vector including all m derivatives of $p(x)$ (see [9]). It is shown in [9] that the first-order optimality condition for the objective function (44) is as follows:

$$p(x_i) - y(x_i) + \rho \, \sigma_i^2 w_i = 0, \quad \forall 1 \le i \le N, \tag{45}$$

where ρ is a parameter proportional to λ. In summary, the coefficient of the regression can be derived by solving:

$$\begin{bmatrix} F & V^T \\ V & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} f(x_i) \\ 0 \end{bmatrix}, \tag{46}$$
$$F_{ij} = \varphi(\|x_i - x_j\|) + \rho \delta_{i,j} \, \sigma_i^2, \qquad V = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix},$$

where $\delta_{i,j}$ is the Kronecker delta.

The problem which is left to solve when computing the regression is to find an appropriate value of $\rho \in [0, \infty)$. Solving (46) for any value of ρ gives a unique regression, denoted $p(x, \rho)$. The parameter ρ is then obtained by a predictive mean-square error criteria developed in §4.4 in [30], which is given by imposing the following condition:

$$T(\rho) = \sum_{i=1}^{N} [\frac{p(x_i, \rho) - y(x_i)}{\sigma_i}]^2 = 1. \tag{47}$$

For $\rho \to \infty$, $w_i \to 0$, and the solution of (46) is a weighted mean-square linear regression, which is obtained by solving (47). If $T(\infty) \le 1$, we take this linear regression as the best current regression for the available data. Otherwise, we have $T(\infty) > 1$ and (by construction) $T(0) = 0$; thus, (47) has a solution with finite $\rho > 0$ which gives the desired regression.

If $T(\infty) > 1$, we thus seek a ρ for which for $T(\rho) = 1$. Following [30], using (46), (47) simplifies to:

$$T(\rho) = \rho \sum_{i=1}^{N} w_{i,\rho} \, \sigma_i^2 = 1, \tag{48}$$

where $w_{i,\rho}$ is the w_i which is obtained by solving (46). Define Dw and Dv as the vectors whose i-th elements are the derivatives of w_i and v_i with respect to ρ, then

$$T'(\rho) = \sum_{i=1}^{N} Dw_i \sigma_i^2 + \rho \sum_{i=1}^{N} Dw_{i,\rho} \sigma_i^2,$$
$$\begin{bmatrix} F & V^T \\ V & 0 \end{bmatrix} \begin{bmatrix} Dw \\ Dv \end{bmatrix} + \begin{bmatrix} \rho\Sigma_2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where Σ_2 is a diagonal matrix whose i-the diagonal element is $\rho\,\sigma_i^2$. Therefore, the analytic expression for the derivative of $T(\rho)$ is available. Thus, (47) can be solved quickly using Newton's method.

The regression process presented here, imposing (48) as suggested by [30], is designed to obtain a regression which is reasonably smooth. However, there is no guaranty that this particular regression satisfies the strictness property required in the present work (see Definition 4). Note, however, that by imposing $\rho = 0$, the regression is made strict for arbitrary small β. Thus, to satisfy strictness for a given finite β, the value of ρ must sometimes be decreased from that which satisfies (48), as necessary.

References.

[1] Hernan P Awad and Peter W Glynn. On an initial transient deletion rule with rigorous theoretical support. In *Proceedings of the 38th conference on Winter simulation*, pages 186–191. Winter Simulation Conference, 2006.

[2] Jan Beran. *Statistics for long-memory processes*, volume 61. CRC Press, 1994.

[3] Jan Beran. Maximum likelihood estimation of the differencing parameter for invertible short and long memory autoregressive integrated moving average models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 659–672, 1995.

[4] Pooriya Beyhaghi and Thomas R Bewley. Delaunay-based derivative-free optimization via global surrogates, part ii: convex constraints. *Journal of Global Optimization*, pages 1–33, 2016.

[5] Pooriya Beyhaghi and Thomas R Bewley. Implementation of cartesian grids to accelerate delaunay-based derivative-free optimization. *Journal of Global Optimization*, 2017. submitted.

[6] Pooriya Beyhaghi, Daniele Cavaglieri, and Thomas Bewley. Delaunay-based derivative-free optimization via global surrogates, part i: linear constraints. *Journal of Global Optimization*, pages 1–52, 2015.

[7] Andrew J Booker, JE Dennis Jr, Paul D Frank, David B Serafini, Virginia Torczon, and Michael W Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999.

[8] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. X-armed bandits. *The Journal of Machine Learning Research*, 12:1655–1695, 2011.

[9] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[10] Raymond M Hicks and Preston A Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[11] Jonathan RM Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

[12] Antony Jameson and Seokkwan Yoon. Lower-upper implicit schemes with multiple grids for the euler equations. *AIAA journal*, 25(7):929–935, 1987.

[13] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2008.

[14] Alison L Marsden, Meng Wang, John E Dennis Jr, and Parviz Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5(2):235–262, 2004.

[15] Alison L Marsden, Meng Wang, John E Dennis Jr, and Parviz Moin. Suppression of vortex-shedding noise via derivative-free shape optimization. *Physics of Fluids*, 16(10):83–86, 2004.

[16] Franco Modigliani and Franz E Hohn. Production planning over time and the nature of the expectation and planning horizon. *Econometrica, Journal of the Econometric Society*, pages 46–66, 1955.

[17] Todd A Oliver, Nicholas Malaya, Rhys Ulerich, and Robert D Moser. Estimating uncertainties in statistics computed from direct numerical simulation. *Physics of Fluids (1994-present)*, 26(3):035101, 2014.

[18] Victor Picheny, David Ginsbourger, Yann Richet, and Gregory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13, 2013.

[19] Carl Edward Rasmussen. *Gaussian processes for machine learning*. Citeseer, 2006.

[20] James Reuther, Antony Jameson, James Farmer, Luigi Martinelli, and David Saunders. *Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation*, volume 96. NASA Ames Research Center, Research Institute for Advanced Computer Science, 1996.

[21] James J Reuther, Antony Jameson, Juan J Alonso, Mark J Rimllnger, and David Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2. *Journal of aircraft*, 36(1):61–74, 1999.

[22] Scott T Salesky, Marcelo Chamecki, and Nelson L Dias. Estimating the random error in eddy-covariance based fluxes and other turbulence statistics: the filtering method. *Boundary-layer meteorology*, 144(1):113–135, 2012.

[23] Matthias Schonlau, William J Welch, and Donald R Jones. A data-analytic approach to bayesian global optimization. In *Department of Statistics and Actuarial Science and The Institute for Improvement in Quality and Productivity, 1997 ASA conference*, 1997.

[24] Suresh P Sethi and Gerald L Thompson. *What is Optimal Control Theory?* Springer, 2000.

[25] Suresh P Sethi, Qing Zhang, and Han-Qin Zhang. *Average-cost control of stochastic manufacturing systems*, volume 54. Springer Science & Business Media, 2005.

[26] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[27] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *Information Theory, IEEE Transactions on*, 58(5):3250–3265, 2012.

[28] C Talnikar, P Blonigan, J Bodart, and Q Wang. Parallel optimization for les. In *Proceedings of the Summer Program*, page 315, 2014.

[29] R Theunissen, A Di Sante, ML Riethmuller, and RA Van den Braembussche. Confidence estimation using dependent circular block bootstrapping: application to the statistical analysis of piv measurements. *Experiments in Fluids*, 44(4):591–

596, 2008.

[30] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[31] Qiqi Wang. Forward and adjoint sensitivity computation of chaotic dynamical systems. *Journal of Computational Physics*, 235:1–13, 2013.

[32] Qiqi Wang, Parviz Moin, and Gianluca Iaccarino. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal on Scientific Computing*, 31(4):2549–2567, 2009.