

## **Research** Article

# A Radial Basis Function-Based Optimization Algorithm with **Regular Simplex Set Geometry in Ellipsoidal Trust-Regions**

Tom Lefebvre (),<sup>1,2</sup> Frederik De Belie (),<sup>1,2</sup> and Guillaume Crevecoeur (),<sup>1,2</sup>

<sup>1</sup>Department of Electromechanical, Systems and Metal Engineering. Faculty of Engineering and Architecture, Ghent University, Ghent, Belgium <sup>2</sup>EEDT Decision and Control, Flanders Make, Lommel, Belgium

Correspondence should be addressed to Tom Lefebvre; tom.lefebvre@ugent.be

Received 1 June 2021; Revised 4 January 2022; Accepted 27 January 2022; Published 2 March 2022

Academic Editor: Giovanni Garcea

Copyright © 2022 Tom Lefebvre et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we investigate two ideas in the context of the interpolation-based optimization paradigm tailored to derivative-free black-box optimization problems. The proposed architecture maintains a radial basis function interpolation model of the actual objective that is managed according to a trust-region globalization scheme. We focus on two distinctive ideas. Firstly, we explore an original sampling strategy to adapt the interpolation set to the new trust region. A better-than-linear interpolation model is guaranteed by maintaining a well-poised supporting subset that pursues a near regular simplex geometry of n + 1 points plus the trust-region center. This strategy improves the geometric distribution of the interpolation points whilst also optimally exploiting the existing interpolation set. On account of the associated minimal interpolation set size, the better-than-linear interpolation model will exhibit curvature, which is a necessary condition for the second idea. Therefore, we explore the generalization of the classic spherical to an ellipsoidal trust-region geometry by matching the contour ellipses with the inverse of the local problem hessian. This strategy is enabled by the certainty of a curved interpolation model and is introduced to accounts for the local output anisotropy of the objective function when generating new interpolation points. Instead of adapting the sampling strategy to an ellipsoid, we carry out the sampling in an affine transformed space. The combination of both methods is validated on a set of multivariate benchmark problems and compared with ORBIT.

## 1. Introduction

We consider the following optimization problem with the objective function  $f: \mathcal{X} \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ . We further assume that f is continuously differentiable, smooth, and bounded from below and that  $\nabla f$  is Lipschitz continuous on the subset  $\mathcal{X}$ . We restrict the minimization to a convex subset  $\mathcal{X} \subset \mathbb{R}^n$ , however, we assume that none of these constraints are active at the minimum so that (1) can be considered unconstrained in practice (the subset  $\mathcal X$  is mainly introduced to emphasize that the search is limited to a range of practical interest. It may also be a means to delimit the search space to the definition domain of the function).

$$\min_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n}f(\mathbf{x}).$$
 (1)

Such problems are common in engineering, and consequently, a large family of search algorithms has been tailored to solve such problems. To motivate the development of derivative-free optimization methods, we premise additional assumptions on the computational tractability of the objective.

As is the case in many engineering design applications, we assume that any evaluation of the objective is associated with the execution of a complex simulation or legacy code. Such evaluations are time-demanding and usually render local derivative information inaccessible. These assumptions rule out the use of the members of the family of gradientbased optimization algorithms that exploit the first- and second-order derivative information to search for points that satisfy the first- and second-order optimality conditions. Under these conditions, it is also intractable to approximate derivative information indirectly using, for example, a finite adifference scheme, given the tight evaluation budget that is available. In short, these assumptions imply that the search algorithm has to rely exclusively on direct evaluations of the objective function to probe the local geometry of the problem. The denominator derivative-free optimization

1.1. Overview of DFO Methods. In this family, there are two main classes to be recognized, which are as follows: stochastic search methods and deterministic search methods. Either can be further subdivided into population-based and evolutionary strategies and direct search methods and interpolation-based search methods, respectively. The distinction between stochastic or deterministic here refers to the property that, excluding the *random* initialization of the algorithm, the execution and convergence history is the same for deterministic algorithms whilst being subject to variation for stochastic search algorithms.

(DFO) refers to any search algorithm that uses only the

direct evaluations of the objective function.

Generally speaking, stochastic search algorithms maintain a belief over the solution space representing possible candidate solutions. This belief is either represented with an actual population or with a parametric distribution, e.g., the multivariate normal distribution. Members of the latter class are denoted as evolutionary strategies (ESs). Stochastic search methods proceed with the evaluation of any new members of the population and use stochastic operations to generate a new population. In the past few decades, large varieties of population-based algorithms have been invented that are inspired by the swarm intelligence exhibited by animal populations. Examples are monarch butterfly optimization [1], earthworm optimization algorithm [2], elephant herding optimization [3], moth search algorithm [4], slime mould algorithm [5], Harris hawks optimization [6], artificial bee colony optimization [7], and grey wolf optimization [8], amongst others, which, in some way or another, are all related to the classic particle swarm optimization algorithm [9]. These algorithms have found applications in structural design optimization mostly and are particularly appealing to find global solutions at the cost of sampling the objective function abundantly. Instead of a population, the class of evolutionary search algorithms maintains a parametric distribution, which is used to spawn a sample set, which, in turn, is evaluated and assessed to infer the next parametric belief. The benefit of these evolutionary strategies is that they may be derived from much more principled arguments. Again, their main appeal is that they solve for a global solution and are particularly well-equipped to treat noisy or multi-global objective functions, where other algorithms, even population-based algorithms, are prone to get stuck in local minimal. Examples popular in the machine learning and reinforcement learning communities are the covariance-matrix adaption evolutionary strategy [10–13], natural evolutionary strategies [14–17], and information geometric optimization [18]. On account of their inherent stochasticity, the execution of stochastic search

methods may be subject to excessive covariance, and their actual performance may differ for each individual run. Basically, they battle noise with noise.

Deterministic search methods eliminate that problem. The subclass of direct search methods encompasses the historical Nelder-Mead simplex method [19] and modern pattern search methods [20], such as mesh adaptive direct search (MADS) methods and related variants [21-24]. These direct methods are characterized by the fact that they try to satisfy the zeroth-order optimality criterion by comparing the ordering of evaluated points but usually do not exploit any directional preferences implied by them. The advantage here is that they will not be misled by false positives or false negatives (e.g., a strong objective slope that suddenly flattens) but consequently can also not exploit, or at least to a lesser extent, true negatives and true positives (e.g., a strong objective slope that actually points into the direction of the minimum). The other subclass of direct DFO methods can be described as interpolation-based optimization (IBO) methods. The basic idea here is to approximate the objective function by fitting an interpolation model to the available objective function evaluations. The hypothesis is that the underlying objective function is subjected to tendencies that are picked up by the interpolation model, which, in turn, can be exploited to explore the potential regions of interest. Within this scope, there are basically two main branches that can be classified based on the fact whether they maintain a global or a local interpolation function. The global branch was founded when the efficient global optimization (EGO) algorithm was first published [25]. Members of this class maintain a so-called acquisition function that can be optimized to determine the next best point to be evaluated and added to the interpolation set. A large family of methods has been developed based on this idea. The other branch of strictly deterministic interpolation-based methods is based on the trust-region framework and will be discussed in the following subsection.

In conclusion, we note that the principal idea of interpolation-based optimization, i.e., replacing the objective function with an interpolation-based proxy, has been subject to cross-fertilization with ideas from the population-based stochastic search family and are commonly referred to as surrogate-assisted stochastic search methods. As opposed to the acquisition function used in EGO and related methods, here, a so-called batch infill criterion is used that can generate a whole set of points instead of a single point at once. Examples are TLRBF [26], RBFBS [27], SATLBO [28], SAGWO [29], DSCPSO-EMM [30], amongst others. Further exposition is out of scope.

1.2. Trust-Region IBO and Contributions. This article focuses on the particular deterministic DFO subclass of trust-regionbased IBO methods, or TRIBO for short [31–34]. As opposed to the global IBO methods surveyed above, TRIBOs target a compact neighborhood in each iteration. Such a strategy limits the required amount of sample points to obtain a reliable surrogate in the compact neighborhood considered during that iteration [35]. A standard gradientbased optimizer can then be used to solve the local subproblem confined to the compact neighborhood.

This analysis allows one to identify three key components. A framework is required to maintain a sequence of compact neighborhoods. Secondly, a strategy should be present to populate the new compact neighborhood with interpolation points. In turn, this interpolation set can be used to determine a new local interpolation model that is sufficiently accurate in the compact neighborhood. In this work, we will make use of the radial basis functions (RBF) interpolation framework [36–38]. The first two components are discussed next.

The trust-region globalization scheme offers a framework to generate a sequence of compact neighborhoods that will be referred to as trust-regions henceforth. When the surrogate satisfies Taylor-like error bounds on its function value and gradient for every iteration, convergence to firstorder optimality can be guaranteed [39]. In the context of interpolation, such error bounds can be established, depending on the geometry of the interpolation set used [40]. The *geometry* of an interpolation set can be understood as the distribution of points with respect to a certain region. The quality of geometry is quantified by a set-dependent geometry metric. An arbitrary set of points is then said to be *well-poised* for interpolation within the specified region if the corresponding geometry measure is sufficiently large.

To guarantee a valid interpolation model suited for optimization within a specified trust region, IBO methods maintain a subset of the available interpolation set for which the geometry constant is maximized. Consequently, the well-poisedness of the subset geometry is assured, and so is the model validity. In every iteration, this constant is maximized by removing points from the interpolation set and generating new ones according to the current size and location of the trust region whilst limiting the overall accumulation of objective evaluations. This delicate mechanism is referred to as interpolation set management.

In [32, 41, 42], such a geometry measure was described using a basis of polynomials. Other authors have described this constant in the function of the QR pivots of the Vandermonde matrix determined by the interpolation points [31, 40]. The mentioned methods consider a subset of n + 1 interpolation points to guarantee a valid model. In any of these cases, the interpolation set management pursues *orthogonality* between n interpolation points relative to the current trust-region center. The center then serves as the (n + 1)<sup>th</sup> interpolation point.

The contribution of this article is twofold. Firstly, in our algorithm, we pursue a subset of n + 2 instead of n + 1 interpolation points, as seen in [31, 32]. An original sampling strategy is devised that is inspired by the volumetric extremity properties of the regular *n*-simplex. This strategy has two benefits. For one, we achieve an improved interpolation set distribution, which is heuristically justified for low-dimensional problems. Then, considering that we use the RBF framework to construct a local interpolation model, it is guaranteed that the surrogate will exhibit curvature.

Secondly, we propose to leverage this curvature information. We, therefore, generalize the traditional spherical trust-regions to ellipsoidal trust regions. The motivation to use an ellipsoidal geometry is again twofold. An ellipsoidal 3

trust-region will adapt to the geometry of the objective function and will, therefore, allow larger steps toward more promising regions compared to using spherical geometry, i.e., the trust-region boundaries will align with the contours of the objective function, and the objective will look like a spherical problem. This idea is already leveraged in the context of evolutionary computation [10, 43]. Secondly, our interpolation set management is input-space oriented, meaning it only takes into account the local distribution of the interpolation points and does not consider the associated output values. The introduction of an ellipsoidal trust-region, whose shape reflects the objective's response behavior, allows to account for the output-space when executing the interpolation set management and will, therefore, benefit interpolation-based metamodeling.

1.3. Outline. To keep this article self-contained, we provide a brief review of the general trust-region framework for surrogate management in Section 2. In Section 3, we discuss the radial basis function interpolation framework and provide details on the implementation. Section 4 motivates our choice for using n + 2 interpolation points and introduces our interpolation set management. In Section 5, we discuss the transition toward ellipsoidal trust regions. In Section 6, we validate our methods empirically on a number of benchmark problems.

#### 2. Trust-Region Methods

In this section, we give a brief review of the traditional trustregion globalization scheme and its applications in IBO. Therefore, we will introduce the so-called class of fully linear surrogate models that generalize the characteristics of the surrogate model used by the trust-region methodology from gradient-based models (such as second-order Taylor models) to more general model types (such as radial basis interpolation models).

2.1. Generic Trust-Region Method. The trust-region framework is a widely-used technique to attain the global convergence of gradient-based optimization algorithms. At every iteration k, a *surrogate* model,  $m_k$ , is considered that approximates the actual objective within a neighborhood of iterate  $\mathbf{x}_k$ . Traditionally, this neighborhood, i.e., the trustregion, is modeled as a sphere.

$$\mathscr{B}(\mathbf{x}_{k},\Delta_{k}) = \left\{ \mathbf{x} \in \mathbb{R}^{n} : \left\| \mathbf{x} - \mathbf{x}_{k} \right\| \leq \Delta_{k} \right\},$$
(2)

where  $\mathbf{x}_k$  is the current iterate and  $\Delta_k$  is referred to as the trust-region radius.

A trial step,  $\mathbf{s}_k$ , is obtained by minimizing the surrogate model,  $m_k$ , in the trust-region,  $\mathscr{B}(\mathbf{x}_k, \Delta_k)$ . This minimization problem is referred to as the trust-region subproblem. The resulting trial step,  $\mathbf{s}_k$ , gives rise to the trivial candidate iterate,  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$ .

$$\mathbf{s}_{k} = \arg\min_{\mathbf{s}\in\mathscr{B}(0,\Delta_{k})} m_{k}(\mathbf{x}_{k} + \mathbf{s}).$$
(3)

In every iteration, the size and position of the trustregion are updated in such a way that the generated sequence of iterates converges. The configuration of the trust-region, which is determined by the couple,  $(\mathbf{x}_k, \Delta_k)$ , is updated according to a fundamental scheme based on the ratio of actual to predicted decrease,  $\rho_k$ . The ratio constitutes a metric for the predictive quality of the surrogate with respect to the current trust-radius [35].

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}.$$
(4)

Based on the value of  $\rho_k$ , the trust region is updated according to the following general rules. If the prediction of the surrogate is good, the radius is increased. In case of a bad prediction, the radius is reduced. For other cases, the current radius is simply maintained.

To regulate the expansion and contraction of the trustregion, one takes values  $0 < \Delta_m \le \Delta_M$ ,  $0 \le \eta_1 < \eta_2 < 1$  and  $0 < \gamma_1 < 1 < \gamma_2$ , where  $\Delta_m$  and  $\Delta_M$  represent the minimal and maximal radii, respectively. The value of  $\Delta_k$  is then updated as follows:

$$\Delta_{k+1} = \begin{cases} \max\{\gamma_1 \Delta_k, \Delta_m\}, \rho_k \le \eta_1 \\ \Delta_k, \quad \eta_1 < \rho_k \le \eta_2 \\ \min\{\gamma_2 \Delta_k, \Delta_M\}, \eta_2 < \rho_k \end{cases}$$
(5)

To decide whether to accept or reject a trial step,  $\mathbf{s}_k$ , we adopt the rules from [35]. If the candidate iterate,  $\mathbf{x}_k + \mathbf{s}_k$ , is not accepted, then the trust-radius shrinks, given that  $\rho_k < 0$ .

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k, f(\mathbf{x}_k + \mathbf{s}_k) < f(\mathbf{x}_k) \\ \mathbf{x}_k, & \text{otherwise} \end{cases}$$
(6)

The generic trust-region algorithm is presented in Algorithm 1.

2.2. Trust-Regions and Fully Linear Models. In the gradientbased trust-region method, the surrogate,  $m_k$ , is provided by a quadratic model of the following form:

$$m_{k}(\mathbf{x}) = f(\mathbf{x}_{k}) + \mathbf{g}_{k}^{\dagger}(\mathbf{x} - \mathbf{x}_{k}),$$

$$+ \frac{1}{2}(\mathbf{x} - \mathbf{x}_{k})^{\mathsf{T}}\mathbf{H}_{k}(\mathbf{x} - \mathbf{x}_{k}),$$
(7)

where  $\mathbf{g}_k$  and  $\mathbf{H}_k$  are the evaluations of the first- and secondorder derivatives of the objective function f in the iterate  $\mathbf{x}_k$ , respectively. According to Taylor's theorem, a region exists within the vicinity of  $\mathbf{x}_k$ , where the assumption that model  $m_k$  is a reliable approximation of f is true. A region of reliable approximation can be defined as a region where the surrogate modeling approximation error, reflected by  $\rho_k$ , is small enough such that the descend in the surrogate model,  $m_k$ , restricted to this region corresponds to the descend of the objective function f. It is apparent that the strength of the trust-region method thus depends on the capacity of the updating rules to maintain a region, where the surrogate modeling approximation error is sufficiently bounded so that  $m_k$  is suited for optimization purposes [44, 45]. From that perspective, the trust-region method can be decomposed into two crucial mechanisms that interact to generate the sequence of iterates. A procedure must be provided that can generate a quadratic approximation model  $m_k$  of the goal function f for any given iterate  $\mathbf{x}_k$ . The updating rules will then maintain a series of regions, where the corresponding and consecutive models,  $m_k$ , are suited to generate the descending sequence of iterates as the solution of subproblem (3). The latter mechanism is in that aspect generic and can be employed to maintain a series of trust-regions given any type of model. Therefore the mechanism extends from quadratic models to the interpolation based and thus derivative-free models that will be used here.

In this work, we are interested in the class of so-called fully linear models. The definition of a fully linear model establishes Taylor-like bounds on arbitrary models, and therefore, it generalizes the notion of a reliable approximation. As a consequence, the trust-region framework can be generalized to include such fully linear models.

We adopt the definition from [39, 40] with  $\hat{\mathcal{X}} = \{ \mathbf{x} \in \mathcal{X} : \mathcal{B}(\mathbf{x}, \Delta_M) \subseteq \mathcal{X} \}.$ 

Definition 1. For given  $\kappa_f > 0$ ,  $\kappa_g > 0$ , and for given  $\mathbf{x}_k \in \hat{\mathcal{X}}$ ,  $\Delta_k \in (0, \Delta_M]$ , defining  $\mathcal{B}_k = \mathcal{B}(\mathbf{x}_k, \Delta_k)$ , surrogate model  $m_k$  is said to be fully linear on  $\mathcal{B}_k$ , if for all  $\mathbf{x} \in \mathcal{B}_k$ ,

$$\left| f\left(\mathbf{x}\right) - m_{k}\left(\mathbf{x}\right) \right| \leq \kappa_{f} \Delta_{k}^{2},$$

$$\left\| \nabla f\left(\mathbf{x}\right) - \nabla m_{k}\left(\mathbf{x}\right) \right\| \leq \kappa_{q} \Delta_{k}.$$
(8)

Let us assume two different procedures that can generate a fully linear surrogate model  $m_k$  on  $\mathscr{B}(\mathbf{x}_k, \Delta_k)$  or that are able to verify whether the surrogate model,  $m_k$ , is fully linear on  $\mathscr{B}(\mathbf{x}_k, \Delta_k)$  and that for any given  $\mathbf{x}_k \in \widehat{\mathscr{X}}, \Delta_k \in (0, \Delta_M]$ . If two such procedures are available, (1) can be modified to construct a trust-region algorithm with fully linear models that will converge to a first-order critical point of the function f [39]. Algorithm 2 presents such a trust-region algorithm using fully linear models.

The main modification is the introduction of the termination criteria,  $\|\nabla m_k(\mathbf{x}_k)\|\varepsilon_g/2$ . This condition is a reformulation of the first-order optimality condition on the objective function,  $\|\nabla f(\mathbf{x}_k)\|\varepsilon_g$ . Assuming that  $m_k$  is fully linear on  $\mathscr{B}(\mathbf{x}_k, \Delta_k^g)$ , we have the following:

$$\begin{aligned} \left\|\nabla f\left(\mathbf{x}_{k}\right)\right\| &\leq \left\|\nabla m_{k}\left(\mathbf{x}_{k}\right)\right\| + \left\|\nabla f\left(\mathbf{x}_{k}\right) - \nabla m_{k}\left(\mathbf{x}_{k}\right)\right\| \\ &\leq \frac{\epsilon_{g}}{2} + \kappa_{g}\Delta_{k}^{g}. \end{aligned} \tag{9}$$

Algorithm 2 assures that upon termination, model  $m_k$ will be fully linear on  $\mathscr{B}(\mathbf{x}_k, \epsilon_g/2\kappa_g)$ . Substituting  $\epsilon_g/2\kappa_g$  for  $\Delta_k^g$  in (9) reveals that this is equivalent to satisfying the firstorder optimality criterion,  $\|\nabla f(\mathbf{x}_k)\| \le \epsilon_g$ . We remark that it is thus fundamental to provide procedures that can guarantee the tightest possible bound on  $\kappa_f$  and  $\kappa_g$ . In Section 4, such procedures are developed, considering that model  $m_k$  is an interpolation model.

2.3. Interpolation Models. When first-order derivative information,  $\mathbf{g}_k$ , is not directly available, it is possible to

(1) choose  $\mathbf{x}_0 \in \mathcal{X} \subset \mathbb{R}^n$ ,  $0 < \Delta_m \le \Delta_0 \le \Delta_M$ ,  $0 \le \eta_1 < \eta_2 < 1$  and  $0 < \gamma_1 < 1 < \gamma_2$ (2) for  $k = 0, 1, \ldots$  until convergence **do** (3) obtain model  $m_k$ (4) solve sub-problem (3) to obtain  $\mathbf{s}_k$ (5) evaluate  $f(\mathbf{x}_k + \mathbf{s}_k)$  and compute  $\rho_k$  through (4) (6) update  $\Delta_{k+1}$  and  $\mathbf{x}_{k+1}$  according to (5) and (6) (7) end for

(8) return  $\mathbf{x}_k$ 

ALGORITHM 1: Generic trust-region algorithm.

```
(1) choose \mathbf{x}_0 \in \mathcal{X} \subset \mathbb{R}^n, 0 < \Delta_m \le \Delta_0 \le \Delta_M, 0 \le \eta_1 < \eta_2 < 1 and 0 < \gamma_1 < 1 < \gamma_2
 (2) for k = 0, 1, \dots do
           construct fully linear model m_k valid in \mathscr{B}(\mathbf{x}_k, \Delta_k)
 (3)
           if \|\nabla m_k(\mathbf{x}_k)\| \le \epsilon_q/2 then
 (4)
               if m_k is fully linear in \mathscr{B}(\mathbf{x}_k, \epsilon_a/2\kappa_a) then
 (5)
 (6)
                    stop
 (7)
               else
 (8)
                    set \mathbf{x}_{k+1} = \mathbf{x}_k and \Delta_{k+1} = \epsilon_g / 2\kappa_g
 (9)
           end if
(10) else
(11)
           solve sub-problem (3) to obtain \mathbf{s}_k
(12)
           evaluate f(\mathbf{x}_k + \mathbf{s}_k) and compute \rho_k through (4)
(13)
           update \Delta_{k+1} and \mathbf{x}_{k+1} according to (5) and (6)
(14)
           end if
(15) end for
(16) return \mathbf{x}_k
```

ALGORITHM 2: Trust-region algorithm with fully linear property.

construct a surrogate model based on strategically generated data, i.e., the direct evaluations of the objective function. A model  $m_k$  denotes an interpolation model if it satisfies the interpolation conditions in (10) for a given input set,  $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_q\}$ , with interpolation points,  $\mathbf{z}_i \in \mathbb{R}^n$ , and a corresponding output set,  $\mathcal{F} = \{f(\mathbf{z}_1), \ldots, f(\mathbf{z}_a)\}$ .

$$m_k(\mathbf{z}_i) = f(\mathbf{z}_i), i = 1, \dots, q.$$
(10)

A quadratic model, as in (7), is capable of modeling the curvature of the underlying function f and allows, therefore, to proof global convergence to second-order critical points [39]. Correspondingly, several DFO methods have been developed that operate with quadratic interpolation models [34, 46, 47]. To determine a quadratic model, the number of interpolation points, q, must either equal or exceed q = 1/2(n + 1)(n + 2), thus requiring at least that number of objective function to limit the number of objective function evaluations.

A linear model can be constructed with as few as n + 1 interpolation points. Algorithms that operate with minimal interpolation sets of q = n + 1 are described by [31, 32, 48]. However, the drawback, in this case, is that a linear model is not capable of representing any curvature present in the objective.

In this work, we impose the additional requirement that the minimal number of interpolation points must equal q = n + 2 because it is then guaranteed that the resulting model can represent a curvature, given a proper interpolation framework (see Section 3). We will refer to such models as *better-than-linear* models, noting that these models classify as fully linear models but are constructed with at least one more interpolation point than the minimum for exact linear models q = n + 1.

2.4. Interpolation-Based Optimization. Anticipating that a procedure is available that can generate an interpolation model,  $m_k$ , that classifies as a fully linear model, an interpolation framework is trivially embedded in Algorithm 2.

In Section 3, we will discuss a multivariate interpolation method that can model curvature and is flexible in the sense that it does not impose strict requirements on the number of interpolation points. Within the context of this framework, it will be shown that an interpolation model can be classified as a fully linear model if the set,  $\mathcal{Z}$ , satisfies conditions on the geometry of the points with respect to the trust-region [40]. The geometry can be understood as the scattering of interpolation points inside a specified region. In Section 4, the geometry of an interpolation set is quantified using a novel geometry metric.

The procedures of verifying and generating fully linear interpolation models thus boil down to verifying whether

the geometric conditions are satisfied with respect to having a fully-linear model for a given interpolation set and to complementing an existing or possibly empty set so that the condition is satisfied by construction. In the context of IBO, these two mechanisms are provided by a single procedure that is referred to as the interpolation set management or the sampling strategy. A sampling strategy for minimal interpolation sets of q = n + 2 is proposed in Section 4.

#### **3. Radial Basis Functions**

Algorithm 2 allows to engage the trust-region globalization scheme whilst substituting interpolation models (10) for the surrogate model,  $m_k$ . It requires a flexible, multivariate interpolation modeling framework that can capture curvature and does not impose strict conditions on the number of interpolation points since we will recycle interpolation points from previous iterations. The radial basis function (RBF) interpolation framework has readily shown its utility within IBO [31–33]. The traditional RBF framework decides arbitrarily on the shape of the modeling RBFs by fixing a shape parameter a priori. Alternatively, one can determine the shape parameter based on the interpolation data. In that case, the RBF framework coincides with universal kriging [36], where the shape parameter is referred to as a hyperparameter. The modeling assumptions of the framework and the determination of the model parameters and hyperparameters are discussed next.

3.1. Interpolation Framework. Consider a given set of interpolation points  $\mathcal{F}$  and corresponding objective function evaluations  $\mathcal{F}$ . The RBF multivariate interpolation framework then presumes a function model of the following form:

$$m(\mathbf{x}) = \sum_{\mathbf{z}_i \in \mathcal{F}} \alpha_i \phi(\|\mathbf{x} - \mathbf{z}_i\|) + p(\mathbf{x}),$$
(11)

where  $\phi: \mathbb{R}_+ \longrightarrow \mathbb{R}$  is an element of the class of RBFs and  $p(\mathbf{x})$  is an element of an *n*-variate polynomial space  $P_d^n$ of at most degree *d*, referred to as the polynomial tail. Let  $\hat{P}_d^n = \left\{ \pi_1(\mathbf{x}), \dots, \pi_{\widehat{d}}(\mathbf{x}) \right\}$  be an *n*-variate basis for  $P_d^n$  with  $\hat{d}$  being the size of this *n*-variate basis. Then, the polynomial tail can be represented by a linear combination of this basis, i.e.,  $p(\mathbf{x}) = \sum_{\pi_i \in \widehat{P}^d} \beta_j \pi_j(\mathbf{x})$ , with  $\beta_j$  being the expansion coefficient. Some popular choices for RBF are listed in Table 1. Notice that the function  $\phi(\|\mathbf{x}\|)$  has isosurfaces coinciding with hyperspheres in  $\mathbb{R}^n$ , i.e., the functions radiate out isotropically.

By introducing vectors,  $\phi(\mathbf{x})$  and  $\alpha$ ,  $\in \mathbb{R}^{q}$  and,  $\pi(\mathbf{x})$  and  $\beta$ ,  $\in \in \mathbb{R}^{d}$ , which entries are given by  $\phi(||\mathbf{x} - \mathbf{z}_{i}||), \alpha_{i}, \pi_{j}(\mathbf{x})$  and  $\beta_{j}$ , respectively-, we can reformulate the RBF model as

$$m(\mathbf{x}) = \sum_{\mathbf{z}_i \in \mathcal{Z}} \alpha_i \phi(\|\mathbf{x} - \mathbf{z}_i\|) + \sum_{\pi_j \in \widehat{P}_d} \beta_j \pi_j(\mathbf{x})$$
  
=  $\phi(\mathbf{x})^\top \cdot \mathbf{a} + \pi(\mathbf{x})^\top \cdot \mathbf{\beta}.$  (12)

3.2. Determination of Model Parameters. The RBF function model (12) is uniquely defined by the parameters  $\alpha$  and  $\beta$ . These parameters can be determined as follows:

Let the Vandermonde matrices  $\Phi \in \mathbb{R}^{q \times q}$  and  $\Pi \in \mathbb{R}^{q \times d}$  be defined as  $\Phi_{ik} = \phi(\|\mathbf{z}_i - \mathbf{z}_k\|)$  and  $\Pi_{ij} = \pi_j(\mathbf{z}_i)$ . The set of q interpolation conditions in (10) can then be written as follows:

$$\begin{bmatrix} \Phi & \Pi \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{f}, \tag{13}$$

where the entries of **f** are defined as the function evaluations contained in  $\mathcal{F}$ .

These conditions do not render unique  $\alpha$  and  $\beta$ . Therefore, d additional conditions are required to complement (13) so that the assembled system matrix is nonsingular. To this end, we require that the model  $m(\mathbf{x})$  is the smoothest model of the form described by (12) that complies with the interpolation condition (10). For RBFs, function smoothness can be defined in relation with the concept of conditional positive definiteness (CPD) [37]. RBFs have the specific property that to any RBF,  $\phi$ , one can associate a value  $d_0$  such that if the maximum degree d of the polynomial tail satisfies  $d \ge d_0$ , the matrix  $\Phi$  will be positive definite with respect to all vectors  $\alpha \in \text{null}(\Pi)$ . Equivalently, we have that if  $d \ge d_0$  and  $\Pi^{\top} \alpha = 0$ , it holds that  $(-1)^{d_0+1} \alpha^{\top} \Phi \alpha > 0$ . In conclusion, when CPD is satisfied, i.e.,  $\alpha^{T} \Phi \alpha > 0$ , it can be shown that the corresponding interpolator is the smoothest function of form (12) satisfying (10). For an elaborate proof and additional details on CPD, we refer to [37].

Formally, the matrix representation of the interpolation condition can be complemented as such by the function smoothness condition to obtain the linear system of equations.

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{\Pi} \\ \mathbf{\Pi}^{\mathsf{T}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\alpha} \\ \mathbf{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}.$$
(14)

The model parameters are determined uniquely if the system matrix is nonsingular. It is easily verified that if  $d \ge d_0$ , it is then sufficient that  $\Pi^{\top}$  has a full column rank. If so, the interpolation set is said to be well-poised for interpolation. In Section 4, how the well-posedness corresponds with the geometrical properties of the points in  $\mathcal{X}$  is discussed. [4]

Based on the previous observation, the explicit solution of the model parameters is as follows:

$$\boldsymbol{\beta} = \left(\boldsymbol{\Pi}^{\top} \boldsymbol{\Phi}^{-1} \boldsymbol{\Pi}\right)^{-1} \boldsymbol{\Pi}^{\top} \boldsymbol{\Phi}^{-1} \mathbf{f},$$
  
$$\boldsymbol{\alpha} = \boldsymbol{\Phi}^{-1} \left(\mathbf{f} - \boldsymbol{\Pi} \boldsymbol{\beta}\right),$$
(15)

This result may also be obtained using the universal kriging (UK) framework. Here, a linear predictor of the form  $m(\mathbf{x}) = \eta(\mathbf{x})^{\mathsf{T}} \mathbf{f}$  is considered. The objective function is modeled as the realization of a polynomial tail and a random process,  $f(\mathbf{x}) = p(\mathbf{x}) + w(\mathbf{x})$ . The process' stochasticity is quantified by a spatial correlation function that models the output correlation as a function of the spatial distance between the input values. The correlation is thus expressed in terms of RBF as  $\operatorname{corr}(w(\mathbf{z}_i)w(\mathbf{z}_i)) = \phi(||\mathbf{z}_i - \mathbf{z}_i||)$ . It can be

TABLE 1: Radial basis functions,  $\gamma > 0$ , where  $d_0$  is the minimal polynomial degree to have CPD.

Туре	$\phi(r)$	$d_0$
Cubic	$r^3$	1
Thin plate spline	$r^2\log(r)$	1
Linear	$\max\{0, 1-\gamma r \}$	0
Multiquadratic	$\sqrt{r^2 + \gamma^2}$	0
Gaussian	$\exp\left(-\gamma r^{\beta} ight)$	-1

shown, e.g., [36, 38], that the optimal linear unbiased predictor,  $\eta$ , is then determined by the following system of equations, where  $\xi$  can be interpreted as a Lagrangian multiplier.

$$\begin{bmatrix} \Phi & \Pi \\ \Pi^{\top} & 0 \end{bmatrix} \cdot \begin{bmatrix} \eta \\ \xi \end{bmatrix} = \begin{bmatrix} \phi \\ \pi \end{bmatrix}.$$
 (16)

For additional details, we refer to appendix A.

3.3. Hyperparameters. The value of the shape parameter  $\gamma$ (see Table 1) is arbitrary and affects the receding radial effect of RBF contribution in (12). Heuristics may be provided, however, they usually lack proper theoretical foundation. A suitable value is mostly determined by trial-and-error. Furthermore, we emphasize that each component of the interpolation points  $\mathbf{z}_i$  contributes equally to the distance metric,  $\|\cdot\|$ , i.e., the model presumes the isotropic dependence of the objective function on all of its variables. Such isotropic behavior can be achieved by normalizing the optimization space by performing a linear transformation of the input space. Nonetheless, such measure cannot account for any local deviations on the global trend. Since we are to construct local interpolation models whose validity is not to stretch beyond the limits of the trust-region boundary, we could redefine this linear transformation locally.

To account for local anisotropy in the principal directions of the input space, one may define the distance metric as (17). The RBFs in (12) are then redefined as  $\phi(||\mathbf{x} - \mathbf{z}_i||_{\gamma})$ . It is clear that in this case, the univariate hyperparameter  $\gamma$  has become redundant. In practical implementations of the UK framework, the determination of hyperparameters  $\gamma_i$  is automated using a maximum log likelihood estimate based on the observations,  $\mathcal{F}$ .

$$\|\mathbf{x}\|_{\gamma}^{2} = \sum_{i=1}^{n} \gamma_{i} x_{i}^{2}.$$
 (17)

We refer to appendix A and [49] for elaborate details.

#### 4. Interpolation set Management

In Section 2, we described a trust-region method with fully linear models. It was demonstrated how a sequence of fully linear models  $m_k$  can be utilized to converge to a first-order critical point of the objective function f. When considering interpolation models satisfying (10), the question whether  $m_k$  is a fully linear model boils down to a proper management of the interpolation set  $\mathcal{X}$  used to construct  $m_k$ . This so-called sampling strategy should maintain a proper geometry of the interpolation set.

In [40], the geometry of an arbitrary interpolation set,  $\mathcal{Z}$ , was quantified by a single set-dependent constant directly related to the conditions in (8) to define a fully linear model. The lemma in [40], however, only applies to interpolation models for which the set size equals q = n + 1 exactly. On this basis, the authors in [31] devised an IBO algorithm, employing RBF interpolation models, given at least n+1interpolation points to construct a fully linear model. By applying a QR pivoting strategy directly related to the measure described by [40], their sampling strategy assures that a subset of n+1 interpolation points (amongst the entire available data set) exhibits the required geometry to guarantee a fully linear interpolation model. Possibly, additional samples need to be generated to have the verification of the full linearity of the interpolation model. We, henceforth, refer to these points to verify that  $m_k$  is fully linear as the affine subset. Additional points from the entire data set are subsequently recycled to enhance the modeling capabilities of the RBF model as long as the well-poisedness of (14) is not corrupted.

Having an interpolation model  $m_k$  representing the curvature requires an affine subset of  $q \ge n + 2$  (see Section 3.2) that is adequately embedded in an interpolation-based trust-region algorithm. In other words, we aim to ensure the convergence of Algorithm 2 to second-order critical points whilst only adding a single point to the affine subset with respect to [31, 32]. Since an estimate of the curvature is now always available, the remark on the RBF hyperparameters in Section 3.3 can be generalized to arbitrary directions. The latter is investigated in Section 5.

In Section 4.1, we introduce a generalized version of the theorem in [40] for interpolation sets with size exceeding n + 1. On that basis, we identify the optimal affine subset geometry that corresponds with n + 2 interpolation points in 4.2. Finally, in 4.3, we propose a procedure capable to maintain an interpolation set with an affine subset of n + 2 points so that the interpolation model,  $m_k$ , is fully linear and exhibits curvature.

4.1. Better-Than-Linear Models. The uniqueness of the RBF interpolation model parameters implied well-poisedness, i.e., full column rank, of the matrix  $\Pi^{T}$ . Thus, the size of the interpolation set must equal or exceed the size of the polynomial basis. Given that we address a minimal interpolation set size n + 2 and thus  $\hat{d} \le n + 2$ , the polynomial tail will be linear or equivalently  $p(\mathbf{x}) \in P_1^n$ . A convenient choice for the basis  $\hat{P}_1^n$  is then clearly  $\pi(\mathbf{x})^{T} = \begin{bmatrix} \mathbf{x}^T & 1 \end{bmatrix}$ . Since we will embed the interpolation framework in a trust-region framework, the current iterate  $\mathbf{x}_k$  is always available for interpolation since  $f(\mathbf{x}_k)$  is evaluated for calculating the

**Z**, matrix  $\Pi^{\mathsf{T}}$  can be written as follows:

$$\mathbf{\Pi}^{\mathsf{T}} = \begin{bmatrix} 0 & \mathbf{z}_1 & \cdots & \mathbf{z}_{q-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{Z} \\ 1 & 1^{\mathsf{T}} \end{bmatrix}.$$
(18)

As mentioned in [40], the conditions (8) were related to the interpolation set geometry, and more specifically, they were related to matrix Z. Here, a generalized lemma is presented, establishing similar Taylor-like conditions for *better-than-linear* models, i.e., when  $q \ge n + 2$ . A derivation is included in appendix B.

Lemma 1. Suppose that m is an RBF model with linear tail and RBF contribution  $\alpha^{\mathsf{T}}\phi$  that satisfies the interpolation condition,  $m(\mathbf{z}_i) = f(\mathbf{z}_i)$ , for the interpolation set  $\mathcal{Z}:=\left\{0,\mathbf{z}_1,\ldots,\mathbf{z}_{q-1}\right\}\subset \mathcal{B}(0,\Delta), \text{ where } q\geq n+2 \text{ and satisfies } \|\mathbf{Z}\|^{01}\Delta\leq\Lambda_Z. \text{ Furthermore, suppose that } f \text{ and } m \text{ are } n+2$ continuously differentiable in the hypersphere  $\mathscr{B}(0,\Delta)$  and that  $\nabla f$  and  $\nabla \phi^{\top} \alpha$  are Lipschitz continuous in  $\mathscr{B}(0, \Delta)$ , with Lipschitz constants  $\gamma_f$  and  $\gamma_{\phi}$ , respectively, then the following two inequalities are satisfied for any  $\mathbf{x} \in \mathscr{B}(0, \Delta)$ :

$$|m(\mathbf{x}) - f(\mathbf{x})| \le \left(\frac{5}{2}\Lambda_Z\sqrt{q-1} + \frac{1}{2}\right)(\gamma_f + \gamma_\phi)\Delta^2$$

$$\|\nabla m(\mathbf{x}) - \nabla f(\mathbf{x})\| \le \frac{5}{2}\Lambda_Z\sqrt{q-1}(\gamma_f + \gamma_\phi)\Delta$$
(19)

These conditions are similar to (8). The lemma provides, therefore, an admittedly conservative estimate of the constants  $\kappa_f$  and  $\kappa_a$ , however, foremost it reveals the crucial role of the interpolation set geometry, which is reflected by the value of  $\|\mathbf{Z}\|$ 

It follows that our interpolation set management must ensure the boundedness of  $\Delta_k^{01} \| \mathbf{Z}_k \|$  from below by the a priori fixed value  $\Lambda_Z^{01}$  at each iteration.

4.2. Optimal Critical Set Geometry. The value of  $\Delta^{01} \|\mathbf{Z}\|$  is understood as a measure for the algebraic well-poisedness of a given set  $\mathscr{Z} \subset \mathscr{B}(0, \Delta)$  within  $\mathscr{B}(0, \Delta)$  [40]. For any given number of interpolation points q, we can define the optimal critical set geometry that corresponds with the largest value of  $\|\mathbf{Z}\|$  corresponding to the tightest bounds on  $\kappa_f$  and  $\kappa_a$ . Hence, assuring the set's well-poisedness reduces to generating an affine subset that resembles the optimal set geometry as closely as possible whilst recycling as many points as possible. This strategy balances the requirement to minimize the total amount of function evaluations without corrupting the model quality. Figure 1 depicts optimal geometries for n = 2 dimensions, maximizing the measure  $\|\mathbf{Z}\|$ for affine subsets of size q = n + 1, q = n + 2, and q = 2n + 1, respectively. The optimal set geometry for q = n + 1

complies with the normalized set  $\Delta^{01} \mathcal{Z}, \{0\}$  being orthonormal with respect to the origin, or equivalently when  $\mathcal Z$ coincides with a standard simplex. This set geometry is pursued by QR pivoting strategies as elaborated in [31] and Lagrange or Newton polynomials-based strategies as elaborated in [32, 40]. The optimal set geometry for q = n + 2corresponds with the regular simplex geometry (plus the origin) [19, 50] and that for q = 2n + 1 with that of an orthoplex (plus the origin) [51]. Orthoplex geometry has been recommended by Powell in [34]. This geometry also determines the fixed search directions in many pattern search optimization methods.

In this context, it is relevant to inspect the spatial distribution of the 2D geometries in Figure 1. The optimal set geometries for q = n + 2 and q = 2n + 1 are *unoriented*: the center of mass of the corresponding sets coincide with the origin. It does not apply for q = n + 1, where the set is biased. Note that the above is valid for arbitrary *n*. To quantify this more rigorously, we introduce the metric,  $\vartheta^*(\mathcal{Z})$ , defined as follows:

$$\vartheta^*(\mathscr{Z}) = \max_{\mathbf{x}\in\mathscr{B}(0,\Delta)} \vartheta(\mathbf{x};\mathscr{Z}) = \frac{1}{q} \sum_{\mathbf{z}_i\in\mathscr{Z}} \|\mathbf{x} - \mathbf{z}_i\|.$$
(20)

Its value is equal to the maximum mean distance that any test point in the trust-region can be removed from the modelling set  $\mathcal{Z}$  (including 0). The value gives an indication for the worst-case contribution that the radial basis term (that is directly related to the distance between the test point and the interpolation points) has compared to the linear tail. In Table 2, we compare  $\vartheta^*(\mathscr{Z})$  values for different optimal set geometries. It can be observed that  $\vartheta^*(\mathscr{Z})$  values can be improved by only adding a single point, referring to the shift from the standard to the regular simplex geometry. The contours of  $\vartheta(\mathbf{x}; \mathcal{Z})$  are presented in Figure 1 that allow one to visualize the regions that are not represented well by the optimal set geometries. For  $n \longrightarrow \infty$ ,  $\vartheta^*(\mathscr{Z})$  tends to  $\sqrt{2}$  for all three given configurations, implying that all are as good, or as worse, for infinite dimensional problems.

The discussion based on  $\vartheta^*(\mathscr{Z})$  provides an additional argument to choose the regular simplex over the configuration in Figure 1(a), certainly for low-dimensional problems.

4.3. Regular Simplex Affine Subsets. According to the previous subsection, the optimal affine subset (without the origin) for q = n + 2 is a regular simplex. 0 remains an element of the affine subset that affects the definition of a regular simplex affine subset based on the n + 1 remaining interpolation points.

Formally, the set of all regular simplices lying on the surface  $\partial \mathscr{B}(0,1)$  is given by (21).  $l_n(\cdot)$  (see appendix C) is a scaling factor depending on the radius of the *n*-sphere (here 1).

$$\mathcal{R}^{n}(0,1) = \left\{ \left\{ \mathbf{s}_{0}, \dots, \mathbf{s}_{n} \right\} | \mathbf{s}_{i} \in \partial \mathcal{B}(0,1), \\ \text{and} \left\| \mathbf{s}_{i} - \mathbf{s}_{j} \right\| = l_{n}(1), i \neq j \right\}.$$

$$(21)$$



FIGURE 1: Optimally poised interpolation sets for varying q and n = 2 next to a contour of  $\vartheta(\mathbf{x}; \mathcal{Z})$ . Note that the center point, 0, is each time incorporated in the interpolation sets. (a) q = n + 1. (b) q = n + 2. (c) q = 2n + 1.

TABLE 2:  $\vartheta(\mathcal{Z})$  for different optimal set geometries contained within  $\mathscr{B}(0, 1)$ .

9	<i>n</i> + 1	<i>n</i> + 2	2 <i>n</i> + 2
$n \ge 2$	$1 + \sqrt{2}\sqrt{n^2 + n\sqrt{n}/n} + 1$	$3 + \sqrt{2}\sqrt{n^2 - n}/n + 2$	$1 + \sqrt{2}\sqrt{n^2 - n\sqrt{n}} + \sqrt{2}\sqrt{n^2 + n\sqrt{n}}/2n + 1$
2	1.5652	1.25	1.2453
3	1.5821	1.2928	1.1552
4	1.5856	1.3165	1.2142
5	1.5844	1.3321	1.2513
$\infty$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$

The geometry of the regular simplex can be related to the matrix norm,  $\|\mathbf{Z}\|$ , by examining the hypervolume spanned by the convex hull of the affine subset,  $\mathcal{Z}$ , {0}. Consider, therefore, the hypervolume,  $V(\mathcal{Z})$ , of an arbitrary set,  $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{n+1}\}$ , defined in (22) with spanning vectors,  $\{\mathbf{z}_2 - \mathbf{z}_1, \ldots, \mathbf{z}_{n+1} - \mathbf{z}_1\}$ . As outlined in [50, 52], the regular simplex has volumetric extremity properties and maximizes the set well-poisedness metric,  $\|\mathbf{Z}\|$ .

$$V(\mathscr{Z}) = \frac{\left|\det\left(\left[\mathbf{z}_{2} - \mathbf{z}_{1} \cdots \mathbf{z}_{n+1} - \mathbf{z}_{1}\right]\right)\right|}{n!}.$$
 (22)

Hence, the algebraic problem of maximizing  $\|\mathbf{Z}\|$  coincides with the geometric problem of maximizing the volume of the convex hull of  $\mathcal{Z}$ . Approaching the problem from this point of view helps to devise an intuitive interpolation set management, resulting in a critical subset that is (close to) an element of the set  $\mathcal{R}^n(\mathbf{x}_k, \Delta_k)$ , whilst recycling the interpolation points was evaluated in previous iterations.

Based on the hypervolume  $V(\mathcal{Z})$  of the subset  $\mathcal{Z}$ , a metric can be construed whether  $\mathcal{Z}$  matches a regular simplex sufficiently. The principal idea is that  $V(\mathcal{Z})$  should be close to the theoretical maximal hypervolume of the regular simplex,  $V^*$ . In particular, we require the hypervolume that is not smaller than the maximal hypervolume multiplied with some predefined relative error factor  $0 \le \mu_1 \le 1$ . If  $V(\mathcal{Z}) \ge \mu_1 V^*$  is not satisfied, new interpolation points need to be generated. This procedure guarantees  $\Delta^{01} \|\mathbf{Z}\| \ge \Delta_2^{01}$ .

Before we elaborate the particularities of our set management, we mention that each set of points  $\mathcal{R} \in \mathcal{R}_m^n(0, 1)$ , is a lower order regular simplex itself, where *m* is the dimension of the regular subsimplex considered, and therefore,  $m \le n$ .  $\mathcal{R}_m^n(0, 1)$  is defined in (23). This statement holds since any subset of a set  $\mathcal{R} \in \mathcal{R}^n(0, 1)$  simply inherits the property that the distance between every two points is the same.

$$\mathcal{R}_{m}^{n}(0,1) = \{\{\mathbf{s}_{0},\ldots,\mathbf{s}_{m}\},\$$

$$Vert \exists \{\mathbf{s}_{m+1},\ldots,\mathbf{s}_{n}\} \colon \{\mathbf{s}_{0},\ldots,\mathbf{s}_{n}\} \in \mathcal{R}^{n}(0,1)\}.$$
(23)

4.3.1. Algorithm Expand2volume. Based on the previous idea, Algorithm 3 constructs an affine subset in a systematic manner so that  $m_k$  classifies as a fully linear model for given  $\Lambda_Z$ . The goal is to attain an affine subset with hypervolume  $V(\mathcal{Z}) \ge \mu_1 V^*$  whilst adding as few points as possible to the available set of interpolation points. The input is the set of all interpolations points available, scaled with respect to the trust radius,  $\Delta_k$ , and unbiased with respect to the center,  $\mathbf{x}_k$ , i.e.,  $\mathcal{Z}_k = \{\mathbf{z}_i = \Delta_k^{01} (\mathbf{s}_i - \mathbf{x}_k) | \mathbf{s}_i \in \mathcal{S}_k, \{\mathbf{x}_k\}\}$ .  $\mathcal{S}_k$  is the set incorporating every point evaluated so far. The current origin is excluded such that we remain with only the candidate interpolation points to form the affine subset.

The algorithm embarks on removing all points that are outside a certain periphery, parametrized by the variable  $\theta_1 \ge 1$ . Note that we artificially enlarge the trust-region since we allow parameter  $\theta_1$  to be greater than 1. We can always assure that the interpolation model is fully linear by redefining  $\kappa_f$  and  $\kappa_g$ . It ensures that in practice, the number of interpolation points actually accepted to be part of the affine subset is slightly increased, improving the overall convergence. Otherwise, the algorithm might end up expanding an empty set too regularly (certainly for larger *n*). If this operation yields a subset,  $\mathcal{Z}_1$ , that is still larger than the required affine subset size of n + 1 (which, in practice, will almost never occur), all points that are furthest away from the trust-surface are additionally removed.

If the size of the resulting subset,  $\mathscr{Z}_1$ , is smaller than n + 1, it is expanded by calling Simexpand Algorithm 4. The procedure described by Simexpand exploits the idea that if the set  $\mathscr{Z}_1$  were to be an element of  $\mathscr{R}_{n-m}^n(0,1)$ , we can expand it with an element of  $\mathscr{R}_{m-1}^n(0,1)$  so that their union would then constitute an element from  $\mathscr{R}_n^n(0,1)$ . The expansion element can be generated by rotating and translating a properly sized lower-order regular simplex.

Our numerical implementation proceeds as follows: a set,  $\mathcal{X}$  is determined to be an element of  $\mathcal{R}^{m-1}(0,1)$ 

embedded in the *n*-dimensional workspace. The direction **e** is determined as the orthogonal projection direction of the origin on the subspace spanned by the vertices of  $\mathcal{Z}_1$ . Subsequently, a rotation matrix  $\mathbf{Q}$  is determined so that the basis spanned by the rotated element  $\mathbf{Q}\mathcal{X}$  is orthogonal to the union of the basis spanned by  $\mathcal{Z}_1$  and the direction **e**. In a final step, the rotated element  $\mathbf{Q}\mathcal{X}$  is scaled according to  $r_m^n(1)$  and translated along the direction **e** over a distance  $d_m^n(1)$ . The factors  $r_m^n(.)$  and  $d_m^n(.)$  are explained in appendix C and will only maximize the volume in the occasional coincidence, where  $\mathscr{Z}_1 \in \mathscr{R}^n_{n-m}(0,1)$ . Nonetheless, this same reasoning can be applied when  $\mathscr{Z}_1 \notin \mathscr{R}_{n-m}^n(0,1)$ . In that case, an additional subproblem must be solved to identify the optimal value of  $\sigma$  to maximize the volume of their union. Here,  $\sigma$  is a scaling factor related to  $r_m^n(.)$  and  $d_m^n$  (.). In the description of the algorithm, we make no distinction between a set  ${\mathcal X}$  and its matrix column vector representation X.

We also use the following operators: operator [.] constructs a matrix containing a set of column vectors that span the space determined by the vertices in  $\mathcal{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_q\}$ , i.e.,  $[\mathcal{Z}] = [\mathbf{z}_1 - \mathbf{z}_0 \dots \mathbf{z}_q - \mathbf{z}_0]$ . The function regsim (n, m)outputs a set of vertices  $\mathcal{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_m\}$  that constitutes a regular *m*-simplex in an *n*-dimensional coordinate space. The matrix function QR(.) constructs orthonormal bases for span N and kernel K, determined by the columns of the input matrix, respectively. The mechanics of Simexpand are illustrated in Figure 2. Each subfigure depicts a set  $\mathcal{Z}_1$  that differs in size from the previous set, the corresponding expansion set  $\mathcal{Z}_2$ , the two bases N, and N<sub>X</sub> and the vector  $\mathbf{e}$ .

If the union  $\mathscr{Z}_1 \cup \mathscr{Z}_2$  does not satisfy the criticality condition  $V(\mathscr{Z}_1 \cup \mathscr{Z}_2) \ge \mu_1 V^*$ , an iterative procedure is initiated, where points are removed from the set  $\mathscr{Z}_1$  systematically. Then, the contracted set can be expanded by applying Simexpand again. To avoid that, expansion points are generated too close to a point that was removed in an earlier iteration. Information from the removed point is transferred to the contracted set  $\mathscr{Z}_1$ . Information about a point that gets removed is stored by transferring it to a memory set  $\mathscr{Z}'$ . The mean of this memory set is used to complement the points in  $\mathscr{Z}_1$  that remain before expanding the set with a new regular m + 1-simplex. In this fashion, the added simplex is generated away from the mass center of the original set  $\mathscr{Z}_1$ .

The algorithm ends after at most n + 1 iterations, i.e., when the entire original set  $\mathcal{Z}_1$  is transferred to the memory set  $\mathcal{Z}'$ . We remark that in this case, the mass center is expanded with an (n-1)-regular simplex that is, however, not necessarily an element of  $\mathcal{R}_{n-1}^n(0, 1)$ . As a consequence, the eventual affine subset needs to not be an element of  $\mathcal{R}^n(0, 1)$ . As this does not ensure that the sufficiency condition is satisfied, the algorithm closes with a final verification of the hypervolume and possible expansion of the set  $\mathcal{Z}_2$ .

4.4. Remarks. With respect to algorithms 3 and 4 described above, we add the following: during the initialization of Expand2-volume, all points that are furthest away from the

trust-surface are removed. It is a heuristic. Alternatively, one might run through a more complex scheme, where every possible n + 1 size set is verified with respect to their hypervolume, and the largest set is selected. These points could also be added to the memory set. However, it is a situation that will almost never occur in practice, partially because the optimization algorithm will guide the iterations to unexplored regions of the parameter space. It is also because the trust region will contract when the sequence of iterates starts to converge. Hence, by proper choice of  $\mu_1$  and  $\theta_1$ , it can be avoided that the complete interpolation set of the previous iteration is within the current periphery.

Furthermore, we remark that the alignment of the kernel associated with the space spanned by the union  $\mathscr{Z}_1 \cup 0$  and the space spanned by the regular simplex  $\mathscr{X}$  is underdetermined. Here, this additional degree of freedom remains unexploited. Alternatively, this degree of freedom could be employed by orienting the simplex that is addedso that the generated points are located furthest away from the available set of interpolation points or such that the added points correspond to a mean minimal predicted objective function value.

#### 5. Ellipsoidal Trust-Regions

The procedures described in the previous two sections can be employed to verify and construct a model,  $m_k$ , that satisfies Definition 1 for a fully linear model in a given trust-region,  $\mathscr{B}(x_k, \Delta_k)$ . Hence, these procedures can be used to embed an interpolation framework in the trust-region algorithm with fully linear models described by Algorithm 2, factually realizing the IBO algorithm.

As for now, the geometry of the interpolation points is considered only with regard to the input space and with respect to the trust-region defined in it. Given a spherical trust-region, our input space-oriented sampling strategy will treat every direction equally in the function of a well-conditioned system matrix **Z**. However, from the perspective of optimization, it might be of interest to account for the geometry of the objective function as well.

The gradient of the objective function will be more sensitive to certain search directions than to others as determined by the local curvature of the objective function. Currently, the set management, however, treats all directions equally (as a result of the spherical shape of the trust-region) and will try to homogenize the distance from the trust-region center and in between the interpolation points. Now, as a result of the local curvature, the corresponding function values may vary either faster or slower than that can be expected when solely considering the local objective gradient. Hence, the geometry of the objective function with respect to the arbitrary choice of the optimization space coordinates is reflected by its local curvature.

Therefore, we propose to execute the set management in a transformed space by introducing an auxiliary optimization parameter,  $\mathbf{u}$ . The goal of this transformation is to obtain a normalized curvature of the objective function in the transformed optimization space. We propose to use an adaptive affine mapping from the transformed space to the



FIGURE 2: Illustration of the expansion mechanism of Simexpand for varying m and n = 3. (a) m = 1, (b) m = 2, and (c) m = 3.

Input: 
$$\mathscr{Z}_0 = \mathscr{Z}/0, \theta_1, \mu_1$$
  
Output:  $\mathscr{Z}_2$   
(1) obtain  $\mathscr{Z}_1$  with elements  $\mathbf{z}_i \in \mathscr{Z}_0$  such that  $\|\mathbf{z}_i\| \le \theta_1$   
(2) define  $\mathscr{Z}_2 = \emptyset$   
(3) if  $|\mathscr{Z}_1| > n + 1$  then  
(4) order  $\mathbf{z}_i$  such that  $|1 - \|\mathbf{z}_1\|| \le \ldots \le |1 - \|\mathbf{z}_q\||$   
(5) obtain  $\mathscr{Z}_1 = \mathscr{Z}_1 / \{\mathbf{z}_{n+2}, \ldots, \mathbf{z}_q\}$   
(6) else if  $|\mathscr{Z}_1| < n + 1$  then  
(7) get  $\mathscr{Z}_2 = \text{Simexpand}(\mathscr{Z}_1)$   
(8) end if  
(9) find initial memory  $z' = \mathbf{z}_i \in \mathscr{Z}_1$ :  $\|\mathbf{z}_i\| \le \|\mathbf{z}_j\|, \forall j$   
(10) define memory set  $\mathscr{Z}' = z'$   
(11) while  $V(\mathscr{Z}_1 \cup \mathscr{Z}_2) < \mu_1 V^*$  or  $|\mathscr{Z}_1| > 0$  do  
(12) find  $\mathbf{z}_i \in \mathscr{Z}_1$ :  $\|\mathbf{z}_i - z'\| \le \|\mathbf{z}_j - z'\|, \forall j$   
(13) contract  $\mathscr{Z}_1 = \mathscr{Z}_1 / \mathbf{z}_i$   
(14) redefine memory set  $\mathscr{Z}' = \mathscr{Z}' \cup \mathbf{z}_i$   
(15) expand  $\mathscr{Z}_2 = \text{Simexpand}(\mathscr{Z}_1 \cup \text{mean}(\mathscr{Z}'))$   
(16) end while  
(17) if  $V(\mathscr{Z}_1 \cup \mathscr{Z}_2) < \mu_1 V^*$  then  
(18)  $\mathscr{Z}_2 = \mathscr{Z}_2 \cup \text{Simexpand}(\mathscr{Z}_2)$   
(19) end if  
(20) return  $\mathscr{Z}_2$ 

ALGORITHM 3: Expand2volume.

original space that changes with the trust-region iterations. Notice that the origin in the auxiliary space corresponds with the trust-region center in the original space.

$$\mathbf{x}(\mathbf{u}) = \mathbf{A}_k \mathbf{z} + \mathbf{b}_k = \mathbf{A}_k \mathbf{u} + \mathbf{x}_k.$$
 (24)

The trust-region will be stretched along those directions, where the local gradient is varying slowly and is shrunk in those directions, where the local gradient is varying rapidly. As a consequence, the objective function values evaluated along the trust-region boundary will increase or decrease in equal amounts with respect to the objective function value at the trust-region center. It will have a positive effect on the hyperparameter tuning described in 3.3 since the principal directions are now homogenized. Another consequence is that a spherical trust-region in the auxiliary space  $\mathbf{u}$  will now have an ellipsoidal shape in the original optimization space illustrated by figures 3(a) and 3(b). As a result, a step of length 1 taken in the auxiliary space represented in the original space may exceed the original spherical trust-region radius. Two situations can be distinguished when inspecting the directional curvature. When having a large directional curvature on the one hand, the ellipsoidal geometry will have a stabilizing effect, considering that the set of admissible steps (i.e., the trust-region) becomes less aggressive in terms of large jumps in the objective function, which will also have an indirect effect on the modelling procedure. On the other hand, when having a small

Input: 
$$\mathscr{T}_{1}$$
  
Output:  $\mathscr{T}_{2}$   
(1)  $m = n - |\mathscr{T}_{1}|$   
(2)  $\mathscr{X} = \operatorname{regsim}(n, m - 1)$   
(3)  $[\mathbf{N}, \sim] = QR([\mathscr{T}_{1}]), [\mathbf{N}_{0}, \mathbf{K}_{0}] = QR([\mathscr{T}_{1} \cup 0]), [\mathbf{N}_{X}, \mathbf{K}_{X}] = QR([\mathscr{X} \cup 0])$   
(4)  $[\sim, \mathbf{e}] = QR([\mathbf{N}_{0}])$   
(5)  $\mathbf{Q} = [\mathbf{K}_{0}\mathbf{N}_{0}] \cdot [\mathbf{N}_{X}\mathbf{K}_{X}]^{\top}$   
(6)  $\sigma^{*} = \arg\min V(\sigma \cdot \mathbf{1}_{m}^{\top} \cdot \mathbf{e} + \sqrt{1 - \sigma^{2}} \cdot \mathbf{Q} \cdot \mathscr{X})$   
(7)  $\mathscr{T}_{2} = [\mathbf{z}_{1}^{\sigma} \dots \mathbf{z}_{n-|\mathscr{T}|}] = \sigma^{*} \cdot \mathbf{1}_{m}^{\top} \cdot \mathbf{e} + \sqrt{1 - \sigma^{*2}} \cdot \mathbf{Q} \cdot \mathscr{X}$   
(8) return  $\mathscr{T}_{2}$ 

ALGORITHM 4: Simexpand.

directional curvature, the ellipsoidal trust region will allow the optimization procedure to take large steps in the direction of a slowly varying gradient. In this manner, the original optimization space will be explored more rapidly. For instance, in figure 3(b), the trust-region in the auxiliary space encloses a larger part of the inner contour that contains the actual minimum. Similar ideas to exploit the curvature information have been used in the context of pattern search algorithms [?], however, to the extent of our knowledge, not to reshape the trust-region geometry in the context of IBO.

5.1. Curvature Normalization. The goal of the coordinate transformation is to normalize the local curvature of the problem in the new coordinate space. This condition can be expressed mathematically as given. For reasons that will become apparent, we introduce an additional scaling factor  $\lambda_k^2$ .

$$\nabla_{\mathbf{u}}^{2} f(\mathbf{x}_{k}) = \mathbf{A}_{k}^{\top} \nabla_{\mathbf{x}}^{2} f(\mathbf{x}_{k}) \mathbf{A}_{k} = \lambda_{k}^{2} \mathbf{I}_{n}.$$
 (25)

Assume that we now have an estimate  $\mathbf{B}_k$  for the Hessian  $\nabla_{\mathbf{x}}^2 f(\mathbf{x}_k)$ . We can formulate an expression for  $\mathbf{A}_k$  by considering the singular value decomposition (SVD) of matrix  $\mathbf{B}_k$ . The SVD decomposition of the positive definite matrix  $\mathbf{B}_k$  is given by  $\mathbf{U}_k \Lambda_k \mathbf{U}_k^{\mathsf{T}}$ . Here,  $\mathbf{U}_k$  is an orthonormal matrix and  $\Lambda_k$  is a diagonal matrix with the singular values  $\sigma_1, \ldots, \sigma_n$  on its main diagonal. It is then easily verified that,

$$\mathbf{A}_k = \lambda_k \mathbf{U}_k \mathbf{\Lambda}_k^{01/2}.$$
 (26)

Finally, we will utilize the scaling factor  $\lambda_k$  to keep subproblem (3) as it manifests in the auxiliary coordinate space proportional to its manifestation in the original coordinate space. Mathematically, this proportionality can be realized by determining a measure that remains invariant under the transformation. Recall that as a result of the affine coordinate transformation, a spherical trust-region in the auxiliary optimization space will adopt an ellipsoidal geometry in the original optimization space. We propose the following measure of proportionality: the value of  $\lambda_k$  is determined such that the spherical and ellipsoidal trust-region representations remain isovolumetric.

Consider that the volume of an *n*-dimensional hypersphere with radius  $\Delta_k$  is equal to  $\eta(n) \cdot \Delta_k^n$  and that the volume of an *n*-dimensional ellipsoid with the given half principal axis lengths,  $a_1, \ldots, a_n$ , is equal to  $\eta(n) \cdot \prod a_i$ , where  $\eta_n$  is a mutual scaling factor. We also have that by the construction of  $\mathbf{A}_k$ , half the principal axis lengths can be written in the function of the singular values as  $a_i = \lambda_k \sigma_i^{01/2}$ . An isovolumetric relation between the trust-region representations in both the original as well as the auxiliary coordinate space can be maintained by proper choice of the scaling factor  $\lambda_k$ , i.e.,

$$\lambda_k = \Delta_k \prod_{i=1}^n \sigma_i^{1/2n}.$$
 (27)

The exact coordinate transformation of (24) can then be written explicitly as follows:

$$\mathbf{x}(\mathbf{u}) = \Delta_k \prod_{i=1}^n \sigma_i^{1/2n} \mathbf{U}_k \mathbf{\Lambda}_k^{01/2} \mathbf{u} + \mathbf{x}_k.$$
 (28)

5.2. Exponential Smoothing. Several methods can be devised to provide an estimate of the Hessian,  $\mathbf{B}_k$ . Since our sample strategy ensures that the local interpolation model is curved, we can calculate an approximation of the Hessian at the next iterate in the auxiliary coordinate space directly from the model, i.e.,  $\nabla_{\mathbf{u}}^2 m_k (\mathbf{u}_{k+1})$ .

This estimate can be calculated back to the original coordinate space to provide an estimate for the actual Hessian,  $\hat{B}_{k+1} = \mathbf{A}_k \nabla_{\mathbf{u}}^2 m_k (\mathbf{u}_{k+1}) \mathbf{A}_k^{\mathsf{T}}$ . To get a smooth approximation, we propose to use exponential smoothing. Note that if  $\mu_2 = 1$ , this framework reduces to the original spherical trust-region framework.

$$\mathbf{B}_{k+1} = \mu_2 \mathbf{B}_k + (1 - \mu_2) \widehat{B}_{k+1}, \mathbf{B}_0 = \mathbf{I}_n.$$
 (29)

#### 6. Numerical Results

In this section, we present the numerical results of the presented derivative-free optimization algorithm applied on a set of unconstrained problems from the CUTEst test environment [53]. As indicated by the neighborhood representation metric,  $\vartheta(\mathcal{Z})$ , the method should be especially suited for low-dimensional problems and was developed from this point of view. We have applied the algorithm on a set of test functions with a maximum problem dimension



FIGURE 3: Illustration of a spherical trust-region and its isovolumetric ellipsoidal counterpart for the two-dimensional Rosenbrock function. The curvature was normalized using the exact Hessian. (a) Spherical trust-region. (b) Ellipsoidal trust-region.

n = 8. All tested algorithms were initialized from the traditional starting point [54] and are run once provided that their execution is otherwise entirely deterministic. The following benchmark procedure has been adopted, and the results are presented in Table 3. The results as obtained with 4 different versions of the present algorithm have been compared with those obtained with its closest competitors in the context of derivative-free optimization, being ORBIT [31] and NMSMAX. Since ORBIT outperforms the NEWUOA algorithm (at least for small scale problems [31]), we did not compare our algorithm with NEWUOA [34]. The details of ORBIT are elaborated in the associated paper and are essentially similar to the details of the presented algorithm when the spherical trust-region framework is used and are apart from the novel regular simplex-based sampling strategy. We used the implementation made available at [55]. The NMSMAX algorithm is a frequently practiced implementation of the Nelder-Mead simplex method [19] and is made available in the matrix computation toolbox [56]. The NMSMAX algorithm was initialized with a regular simplex in accordance with the present algorithm and is included as a benchmark.

Each algorithm was given a budget of 1000 function evaluations. Since every test function has a known global minimum,  $f^*$ , the tolerance of having a convergence of an algorithm was  $10^{-6}$ , i.e.,  $f(\mathbf{x}_k) - f^* < 10^{-6}$ . The internal stopping conditions of each of the algorithms were set to 0 so that an algorithm only stops when this benchmark condition was satisfied or when it exceeded its evaluation budget. These internal stopping conditions include the gradient condition (9) and the minimal simplex size for NMSMAX. Only the total number of function evaluations and the difference between the final objective function value and the known minimum are recorded. We did not consider the CPU time and performance and data profiles that are gaining currency in the optimization community [57, 58]. Since the focus of the paper is primarily on the novelty of the sampling strategy and the ellipsoidal trust-region framework, we reason that the adopted approach was best-suited to illustrate the elaborated methods.

Each algorithm was applied using its standard settings to mimic the realistic practical usage. The only parameter that was varied was the initial trust-region radius, or in case of the NMSMAX algorithm, the size of the initial simplex. It corresponds with an arbitrary uniform scaling of the original optimization space, which is a standard practice when performing optimization routines. The standard parameter settings of the presented algorithm are given in Table 4, those of each of the competing algorithms can be found in the respective references. The parameters present in Table 4 were chosen via trial-and-error on a limited benchmark set. Those corresponding to the ORBIT algorithm were adopted from that reference. For future development, a short description of the influence of each parameter is included in table as well.

We have considered 4 different versions of the present algorithm. Details about the numerical implementation are given in the next subsection. These 4 versions allow us to study the separate effects of the two presented mechanisms, namely the novel regular simplex-based sampling strategy and the ellipsoidal trust-region mechanism. The different versions are defined in Table 5 and are related to the values of  $\mu_1$  and  $\mu_2$  from Algorithm 3 and (29), respectively. To study the effects of the regular simplex-based sampling strategy apart from the ellipsoidal trust-region framework, version V1 can be compared with the ORBIT and NMSMAX algorithm. Note that in this case, the interpolation set management is purely input space-oriented. To study the effect of the ellipsoidal trust-region framework, version V2 can be compared with respect to V1. Recall that in this case, the shape of the trust-region is adapted to the output behavior of the objective function. To study the effect of the novel sampling strategy in combination with the ellipsoidal trustregion, versions V2 to V4 can be compared. Note that when parameter  $\mu_2$  (29) is equal to 1, it corresponds with a spherical trust-region framework. It is not possible to study the behavior of the ellipsoidal trust-region framework in combination with the sampling strategy of ORBIT, given that this strategy does not assure that a better-than-linear model is available every iteration.

TABLE 3: Comparative results between 4 versions of the algorithms ORBIT and NMSMAX.

Nama		Total number of function evaluations					Absolute difference to known minimum						
Name n		V1	V2	V3	V4	ORBIT	NMS	V1	V2	V3	V4	ORBIT	NMS
BEALE	2	28	25	32	42	419	63	5,75e-07	6,25e-09	3,10e-07	1,34e-07	5,16e-10	2,93e-07
BRANIN	2	24	25	32	36	53	56	1,10e-07	5,35e-07	7,60e-08	6,14e-07	1,10e-05	7,81e-07
ROSENBROCK	2	37	37	33	39	838	100	2,51e-07	9,56e-08	9,40e-07	1,69e-07	1,80e-03	3,71e-07
SCHWEFEL36	2	26	25	26	32	69	69	9,35e-07	4,29e-07	1,44e-07	7,43e-08	9,17e-07	4,39e-07
GULF	3	494	159	165	226	1000	389	3,29e-07	7,71e-07	1,84e-09	8,95e-07	2,66e+00	5,60e-07
HART3	3	52	46	49	69	348	129	9,98e-09	8,33e-07	7,14e-07	2,64e-08	3,91e-06	6,08e-07
HELIX	3	53	42	48	48	443	115	8,29e-07	5,97e-07	3,08e-07	4,50e-07	4,61e-04	4,31e-07
ROSENBROCK	3	77	68	80	88	1000	176	3,81e-07	4,41e-07	6,66e-07	1,18e-07	5,20e-03	5,12e-07
SCHWEFEL4	3	55	46	41	67	77	93	1,72e-07	7,98e-08	7,20e-07	1,81e-08	8,05e-07	7,07e-07
BROWNDEN	4	118	84	112	121	398	236	6,77e-07	9,34e-06	5,16e-06	3,44e-06	9,60e-06	9,77e-06
HART4	4	50	42	48	38	62	122	3,86e-07	2,78e-07	3,95e-07	2,94e-07	5,56e-07	8,85e-07
POWELL	4	201	92	104	148	615	366	6,77e-07	7,43e-07	8,85e-07	4,51e-07	6,01e-05	9,45e-07
ROSENBROCK	4	136	111	143	144	1000	332	8,67e-07	9,90e-07	3,10e-07	7,74e-07	2,75e-02	8,78e-07
SCHWEFEL4	4	63	61	62	74	112	156	2,46e-07	9,04e-07	5,55e-07	3,18e-07	9,25e-07	9,75e-07
WOOD	4	166	79	85	109	255	217	6,35e-07	5,16e-07	3,78e-07	2,13e-08	4,04e-04	7,57e-07
BIGGS	6	660	294	243	529	156	1000	7,54e-04	1,52e-07	9,65e-07	9,48e-07	6,60e-03	5,68e-05
HART6	6	104	116	95	107	139	179	5,63e-07	5,43e-07	5,23e-07	4,08e-07	8,21e-07	9,39e-07
ROSENBROCK	6	302	257	264	435	1000	818	3,02e-07	6,78e-07	8,09e-07	7,34e-07	7,22e-02	9,90e-07
SCHWEFEL4	6	124	116	111	169	173	221	8,37e-07	7,10e-07	2,38e-07	7,37e-07	9,47e-07	8,81e-07
TRID	6	405	206	205	194	399	541	6,30e-03	6,67e-07	2,77e-07	1,79e-07	6,90e-03	7,94e-07
WATSON	6	117	83	97	119	222	305	2,11e-07	8,63e-07	1,56e-07	4,16e-07	9,83e-07	7,21e-07
POWELL	8	168	135	139	151	225	410	7,61e-07	2,88e-07	5,77e-07	3,95e-07	1,00e-06	9,70e-07
ROSENBROCK	8	912	648	624	769	1000	1000	7,41e-07	7,25e-07	9,35e-07	8,96e-07	4,81e-01	2,75e-01
TRID	8	1000	648	585	778	1000	1000	1,10e-04	9,81e-07	8,49e-07	3,68e-07	3,53e-04	2,40e-03

TABLE 4: Standard settings of the algorithm.

		Determines the set of points that is considered in algorithm 3. For $\theta_1 = 1$ , all points must be inside the trust-region. For slightly
$ heta_1$	1.25	larger values, the nearby points are also considered. Large values will positively bias the volume so that even for poor set
		distributions, the volumetric condition will be satisfied.
$\eta_1$	0	Standard trust-region parameter. Adopted from [35].
$\eta_2$	0.6	See $\eta_1$
$\gamma_1$	0.5	See $\eta_1$
$\gamma_2$	2	See $\eta_1$
ß	$10^{-2}$	Lower-limit for the hyperparameters in the RBF framework. Determines how fast the RBFs decay. For normalized search space
$P_1$	10	and reasonably smooth objective functions, it was determined to be a good limit by trial-and-error.
$\beta_2$	$10^{-1}$	Upper-limit for the hyperparameters in the RBF framework. See $\beta_1$ .
$\Delta_m$	0	Minimal trust-region size. Determines the smallest focus the trust-region framework can have.
$\Delta_M$	$3\Delta_0$	Maximum trust-region size. Determines the largest focus the trust-region framework can have. Value was adopted from [31].
		Maximum number of points to be included in the RBF model. The value was chosen arbitrarily and worked well for relatively
0	a 100	low-dimensional problems. When the number is too large, the influence of far-away points may distort the local geometry of the
$q_M$ IC	100	objective. Furthermore, the fitting and evaluating of the interpolation model becomes computationally challenging as a result of
		hyperparameter optimization.

6.1. Numerical Implementation. All described procedures were implemented in MATLAB R2017 computational environment and run on a Dell Laptop with an Intel i7 CPU with 4 cores. All code is available at the GitHub repository in [59]. Some of the subproblems were solved using the available implementations of routine in MATLAB. These are documented next.

As mentioned in Section 4, once an affine subset was determined using Algorithm 3, the available set of interpolation points was reconsidered to enhance the modelling capabilities of the surrogate model. Additional points were added as long as they did not corrupt the well-poisedness of (14) and if the total amount of interpolation points did not exceed the limit  $q_M$ . A similar procedure is performed by ORBIT and was adopted from that source code.

The interpolation framework used here is that of universal kriging. We employed the DACE toolbox to obtain the modelling parameters as described in Section 3. This framework requires the user to define the lower and upper limits of the hyperparameter  $\beta$ ,  $\beta_1$ , and  $\beta_2$ , respectively. More specifically, we used Gaussian RBFs for the following reasons: they generally perform well and exhibit nice interpolation properties that result in a reasonably smooth optimization surface. Other kernels could be considered as well, however, their influence should be limited.

TABLE 5: Tested versions of the algorithm.

$m_1$		0.5	0.75	1
$m_2$	0.95	V2	V3	V4
	1	V1		

Subproblem (3) was solved using the MATLAB function fminsearchcon, an extension of the function fminsearch made available on MathWorks File exchange [60]. Since absolute CPU time was not considered, the solution procedure was executed until fully converged.

#### 6.2. Discussion of Results

6.2.1. Benchmark Problems. By assessing the results displayed in Table  $3_2$  one can observe that, on the whole, the proposed algorithm outperforms both NMSMAX and ORBIT, disregarding some coincidental draws.

Considering that V1 outperformed ORBIT in all but one occasion, we can conclude that the regular simplex-based sampling strategy offers a significant advantage. One might, however, argue that the increased implementational and computational complexity of Algorithm 3 does not justify the advantageous optimization setting in case that less challenging optimization problems are considered. In none of the benchmark problems did V1 outperform V2, V3, or V4, indicating that engaging an ellipsoidal trust-region framework is clearly a beneficial endeavor regardless, given the low implementational cost. However, the downside is that to obtain a curved model, a minimum of n + 2 points are required, making the advanced regular simplex sampling strategy indispensable.

Whether V2, V3, or V4 outperformed one another depends on the objective function at hand. However, on the whole, we could state that V2 is the better candidate for lower-dimensional problems, whilst V3 is the better candidate for higher dimensional problems, as will be confirmed in the next section. Also, considering the convergence history of the different algorithms, it was noted that the ORBIT algorithm occasionally stagnated on a local plateau of the objective function. Since this behavior was also exhibited by V1 (e.g., GULF, TRID; although V1 recovered each time), it demonstrates that compensating for the output behavior of the objective function by the input space-oriented sampling strategy and by altering the shape of the trust-regions is quintessential to recover from such apparent plateaus.

6.2.2. Extended Powell Singular and Extended Rosenbrock. To demonstrate and comment on the mechanics of the elaborated methods in greater detail, convergence results on the 8-dimensional extended Powell singular and extended Rosenbrock are presented in Figure 4 and discussed below.

Consider the convergence history for the extended Powell singular function in Figure 4(a). We will initially discuss the history of the algorithms V1, ORBIT, and NMSMAX. All exhibit approximately the same convergence behavior during the initial stage of the optimization (up to 100 function evaluations). After 100 function evaluations,

the NMSMAX algorithm clearly starts to diverge from this trend. It is because of the fact that the NMSMAX algorithm is bounded to operate on its vertices. At the common point of approximately 100 function evaluations, it reaches a region where the simplex size needs to shrink sufficiently to attain a size that is small enough to accurately grasp the local curvature of the objective function. A similar phenomenon is observed for the ORBIT algorithm, which starts to diverge from the global trend approximately after 200 function evaluations. The phenomenon occurs later for the ORBIT algorithm since it is not bounded to operate on the vertices of its standard simplex, and the solution of (3) is only limited to the convex set,  $\mathscr{B}(\mathbf{x}_k, \Delta_k)$ . Therefore, it can maintain a larger trust-region throughout the iterations than NMSMAX. That algorithm V1 can maintain a larger trustregion longer than ORBIT, and NMSMAX is a result of the improved geometry of the interpolation points. Apart from that, algorithm V1 too fails to reach the benchmark condition within the admissible budget. Now, consider the behavior of algorithm V2. The sampling strategy is equivalent to that of V1, however, the modelling space is now normalized with respect to the estimated curvature of the objective function. It corresponds with an ellipsoidal trustregion geometry in the original optimization space. The effect is hardly observable during the initial stage but clearly demonstrates its usefulness after approximately 500 function evaluations, where V2 diverges from V1 and successfully reaches the benchmark condition. It is argued that because of the transformation, the modeling and optimization steps of the algorithm are now executed in a space, where the objective function behaves more as if it was a quadratic function. The effect of the parameter  $\mu_1$  can be assessed by comparing V2, V3, and V4. The parameter determines how strictly the volume of an exact regular simplex has to be respected. It results in a larger number of interpolation points to be added in every iteration. Therefore, algorithm V3 and V4 diverge from the common trend at about the same moment V1 and ORBIT do. In this region of the optimization space, the size of the trust-region (simplex) starts to shrink for all algorithms. It means that less function evaluations can be recycled from one trust-region to another. Algorithms V3 and V4 set a stricter limit on the volume of the simplex and are, therefore, forced to add more points to complement the recycled set. As a result, they consume more function evaluations than V1 and V2 to cross the same region. Nonetheless, this increase in function evaluations benefits the overall modelling procedure, and V2 is surpassed by V3 eventually.

These observations are confirmed by the convergence history for the extended Rosenbrock function, see Figure 4(b). Here, the results are directly affected by the local behavior of the objective function and all algorithms start to diverge at the starting point. In this case, V1 does reach the



FIGURE 4: Convergence history represented by the evolution of the best relative function value in the function of the total number of function evaluations. (a) Extended Powell Singular, n = 8. (b) Extended Rosenbrock, n = 8.

benchmark condition, again illustrating the improved scattering of points with respect to state-of-the-art. We argue this is a result of the regular simplex based sampling strategy. The beneficial effect of the ellipsoidal trust-region framework is again clearly present as V2 to V4 converge faster than V1. In conclusion, one can observe that the same effect is present for V3 as was the case for the extended Powell singular function. Function evaluations are consumed at a higher rate in the initial stage. This phenomenon ultimately benefits the modelling procedure in the final optimization stage, resulting in a steeper convergence history and even the surpassing of V2 with respect to convergence speed.

#### 7. Conclusion

This paper is about reducing the number of objective function evaluations in derivative-free numerical optimization with surrogate models. Here, the radial basis functions are considered to be surrogate models with the interpolation framework provided by universal kriging. In each iteration of an interpolation-based optimization algorithm, a trust region can be used, wherein an optimization is performed using the surrogate model. The interpolation framework consists of using an interpolation set, of which, a subset of interpolation points needs to be well-poised so that the quality of the surrogate model is maximal. Interpolation set management aims at maximizing this range of model validity.

Currently, at least q = n + 1 interpolation points are considered to guarantee a valid surrogate model. We advanced on interpolation-based optimization by relying on a subset of at least q = n + 2 interpolation points within the interpolation set management to improve the scattering of the interpolation points, but more importantly, to have surrogate models that exhibit curvature. For that purpose, we defined fully linear surrogate models in Definition 1 that can be incorporated in the trust-region Algorithm 2. We implemented RBF models using a flexible multivariate interpolation framework for which universal kriging was used (see Section 3). In Lemma 1, we provide for a given RBF model with linear tail, information on the boundedness of the tolerance of the RBF model, objective function, and information on their gradients, i.e., constants  $K_f$  and  $K_a$  in Definition 1. In this manner, we are able to predict whether an interpolation model can be classified as being fully linear. Since we consider q = n + 2 interpolation points that represent a regular simplex (see Section 4.3), we propose in algorithms 3 and 4 how the interpolation set management needs to be adapted, i.e., how the input points that need to be evaluated in the objective function need to be chosen.

By leveraging on the curvature of the surrogate model that is valid in the presented interpolation set management, we introduced an ellipsoidal trust-region framework that allows morphed trust-regions that adapt to the objective function geometry. This adaptation is implemented by means of curvature normalization with coordinate transformation (24) and filtered Hessian approximation (29). In this manner, we allow to incorporate the output behavior of the objective function in the shape of the trust-region.

The proposed interpolation set management with proper input of interpolation points that follow a well-poised geometry, together with the output behavior of the objective function that is incorporated in the ellipsoidal trust-region, have shown their benefits with respect to the reduction of the number of objective function evaluations when performing elaborate numerical optimizations on various test functions with  $n \le 8$ . To show the impact of internal parameters of the algorithms, i.e.,  $\mu_1$  in Algorithm 3, which is related to the input set, and  $\mu_2$  in (29), which is related to the ellipsoidal trust-region shape, the results of 4 different versions of the presented algorithm are shown in Figure 4 and Table 3. The results clearly illustrate the impact of the used input interpolation points and the ellipsoidal trust region. Moreover, the convergence histories show faster convergence compared to algorithms incorporating at least n + 1 interpolation points and compared to those that use a spherical trustregion [61].

#### Appendix

## A. Universal Kriging

The universal kriging (UK) framework obtains an interpolating approximation model for the function  $f: \mathcal{X} \subset \mathbb{R}^n \longrightarrow f(\mathcal{X}) \subset \mathbb{R}$  as the most likely unbiased linear predictor,  $m(\mathbf{x})$ , based on a set of q interpolation points  $\mathcal{X} = \{\mathbf{z}_1, \dots, \mathbf{z}_q\}, \mathbf{z}_i \in \mathcal{X}$  and a set of q function evaluations,  $\mathcal{F} = \{f(\mathbf{z}_1), \dots, f(\mathbf{z}_q)\}$ . The framework presumes that the modelled function f can be decomposed as follows:

$$f(\mathbf{x}) = p(\mathbf{x}) + w(\mathbf{x}) = \pi(\mathbf{x})^{\top} \boldsymbol{\beta} + w(\mathbf{x}).$$
(A.1)

The function  $p(\mathbf{x})$  is assumed to be an element of a linear function space, A. In the UK framework, one is interested in the specific case, where  $\mathcal{A}$  is the *n*-variate polynomial space,  $P_d^n$ , of at most degree d spanned by the arbitrary basis,  $\hat{P}_d^n = \left\{ \pi_1(\mathbf{x}), \dots, \pi_{\hat{d}}(\mathbf{x}) \right\}.$  We further introduce the vector,  $\pi(\mathbf{x})$ , whose entries are determined by  $\pi_i(\mathbf{x})$  so that we can write  $p(\mathbf{x}) = \pi(\mathbf{x})^{\mathsf{T}}\beta$ , where  $\beta \in \mathbb{R}^d$  is a unique coefficient vector. It is further assumed that  $w(\mathbf{x})$  is an unbiased random process and that the output values  $w(\mathbf{x})$  and  $w(\mathbf{z})$ are correlated. Their correlation is modeled by a function from the class of radial basis functions, i.e.,  $\operatorname{corr}(w(\mathbf{x}), w(\mathbf{z})) = \phi(\|\mathbf{x} - \mathbf{z}\|)$ . One looks for an unbiased linear predictor  $m(\mathbf{x}) = \eta(\mathbf{x})^{\mathsf{T}} \mathbf{f}$ , where the elements of  $\mathcal{F}$ determine the entries of  $\mathbf{f}$ . The model decomposition can be applied to this vector  $\mathbf{f}$ 

The model decomposition can be applied to this vector **f** so that one obtains  $\mathbf{f} = \Pi \boldsymbol{\beta} + \mathbf{w}$ . Here, the entries of  $\mathbf{w}$  are defined as  $w(\mathbf{x}_i) = f(\mathbf{x}_i) - p(\mathbf{x}_i)$  and the matrix  $\Pi \in \mathbb{R}^{q \times d}$  as  $\Pi_{ij} = \pi_j(\mathbf{x}_i)$ . Substitution in the linear predictor yields  $m(\mathbf{x}) = \eta(\mathbf{x})^\top \Pi \boldsymbol{\beta} + \eta(\mathbf{x})^\top \mathbf{w}$ . Hence, we can define an error function between the linear predictor and the true function as follows: $\Delta(\mathbf{x}) = m(\mathbf{x}) - f(\mathbf{x}) = (\eta(\mathbf{x})^\top \Pi - \pi(\mathbf{x})^\top)\boldsymbol{\beta} + (\eta(\mathbf{x})^\top \mathbf{w} - w(\mathbf{x}))$ .

We can now impose two additional properties on the linear predictor that affect the error function  $\Delta$  and that will render the predictor unique. Firstly, the linear predictor should be unbiased, which can be expressed as  $E[\Delta] \equiv 0$ . It follows that  $\Pi^{\top} \eta = \pi$ , and consequently, it follows that  $\Delta = \eta^{\mathsf{T}} \mathbf{w} - w$ . Secondly, we wish to determine  $\eta$  so that the expected squared error is minimized. Given that  $E[\Delta] = 0$ , the expected squared error can be defined as  $E[\Delta^2]$ , and after substitution of the error function, we obtain  $E[(w - \eta^T w)^2]$ . The evaluation of the expectation operator determines that the error is proportional to the functional

 $\epsilon[\eta] = \eta^{\top} \Phi \eta - 2\eta^{\top} \phi + 1$ , where matrix  $\Phi \in \mathbb{R}^{q \times q}$  is defined as  $\Phi_{ij} = \phi(\|\mathbf{z}_i - \mathbf{z}_j\|)$ , and the entries of  $\phi(\mathbf{x})$  are determined by  $\phi(\|\mathbf{x} - \mathbf{z}_i\|)$ . The most likely unbiased linear predictor  $\eta$  is thus the solution of the equality constrained variational problem.

$$\min_{\boldsymbol{\eta},\boldsymbol{\xi}} \frac{1}{2} (\boldsymbol{\eta}^{\top} \boldsymbol{\Phi} \boldsymbol{\eta} - 2\boldsymbol{\eta}^{\top} \boldsymbol{\phi} + 1) + \boldsymbol{\xi}^{\top} (\boldsymbol{\Pi}^{\top} \boldsymbol{\eta} - \boldsymbol{\pi}).$$
(A.2)

The expression of the first-order optimality condition yields the system of linear equations given in (16).

According to the probabilistic interpretation of the random process  $w(\mathbf{x})$ , the function values in  $\mathbf{f}$  can be associated with the outcome of a stochastic experiment. Consequently, it is possible to associate a probability value to the outcome of this experiment. If we make the additional assumption that the random process  $w(\mathbf{x})$  itself behaves as a Gaussian with the expectation  $\mu = \Pi\beta$  and covariance  $\sigma$ , the probability of the experiment is given by,

$$P(\sigma^{2}) = (2\pi\sigma^{2})^{-n/2} \det(\mathbf{\Phi})^{-1/2} \times \dots$$

$$\cdot \exp\left(-\frac{1}{2}\sigma^{-2} \left(\mathbf{f} - \mathbf{\mu}\right)^{\mathsf{T}} \mathbf{\Phi}^{-1} \left(\mathbf{f} - \mathbf{\mu}\right)\right).$$
(A.3)

A maximum log likelihood estimation for the parameter  $\sigma^2$  results in the following estimate:  $\hat{\sigma}^2 = 1/n$  $(\mathbf{f} - \Pi\beta)^{\top} \Phi^{-1} (\mathbf{f} - \Pi\beta)$ . Now, assume that the correlation model corr( $\mathbf{x}, \mathbf{z}$ ) still depends on the additional hyperparameter, say  $\theta$  (e.g.,  $\phi(||\mathbf{x} - \mathbf{z}||; \theta)$ ). Then, the optimal parameter  $\theta^*$  can also be determined by maximizing the likelihood.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} P(\hat{\sigma}^2). \tag{A.4}$$

## B. Error Bounds on Better-than-Linear RBF Models

Using the Einstein summation convention, we can write an RBF model with linear polynomial tail as presented in (B.1). Here, we have implicitly defined the error functions  $e_f(\mathbf{x})$  and  $\mathbf{e}_{\nabla f}(\mathbf{x})$ . We are interested in the bounds on the values these functions can take, given that the interpolation set,  $\mathcal{Z}$ , contains the origin and that both  $\mathbf{x}$  and  $\mathcal{Z}$  are contained within the ball  $\mathscr{B}(0, \Delta)$ .

$$m(\mathbf{x}) \doteq \gamma + \mathbf{x}^{\top} \boldsymbol{\beta} + \alpha_j \phi_j(\mathbf{x}) = f(\mathbf{x}) + e_f(\mathbf{x}),$$
  

$$\nabla m(\mathbf{x}) \doteq \boldsymbol{\beta} + \alpha_j \nabla \phi_j(\mathbf{x}) = \nabla f(\mathbf{x}) + \mathbf{e}_{\nabla f}(\mathbf{x}).$$
(B.1)

Subtracting the left- and right-hand side expressions for  $m(\mathbf{x})$  from the left- and right-hand side expressions of each of the q > n + 1 interpolation conditions,  $\gamma + \mathbf{x}^{\top} \boldsymbol{\beta} + \alpha_j \phi_j(\mathbf{x}) = f(\mathbf{x}_i)$ , one obtains the following set of equalities.

$$\begin{cases} -\mathbf{x}^{\mathsf{T}}\boldsymbol{\beta} = f(0) - f(\mathbf{x}) - \alpha_{j}(\phi_{j}(0) - \phi_{j}(\mathbf{x})) - e_{f}(\mathbf{x}) \\ (\mathbf{z}_{i} - \mathbf{x})^{\mathsf{T}}\boldsymbol{\beta} = f(\mathbf{z}_{i}) - f(\mathbf{x}) - \alpha_{j}(\phi_{j}(\mathbf{z}_{i}) - \phi_{j}(\mathbf{x})) - e_{f}(\mathbf{x})' \end{cases}$$
(B.2)

These inequalities can be manipulated so to relate the error on the function value,  $e_f(\mathbf{x})$ , to the error on the derivative value,  $\mathbf{e}_{\nabla f}(\mathbf{x})$ . For that purpose, we shall first rewrite the functions  $f(\mathbf{x})$  and  $\phi_k(\mathbf{x})$  using the integral representation  $f(\mathbf{x}) = f(\mathbf{z}_i)$  $+ \int_{\mathbf{z}_i}^{\mathbf{x}} \nabla f(\mathbf{s})^{\mathsf{T}} d\mathbf{s}$ . After substituting  $\mathbf{x}(1-t) + \mathbf{z}_i t$  for  $\mathbf{s}$ , it follows that  $\int_{\mathbf{z}_i}^{\mathbf{x}} \nabla f(\mathbf{s})^{\mathsf{T}} d\mathbf{s} = - \int_0^1 \nabla f(\mathbf{x} - t(\mathbf{z}_i - \mathbf{x}))^{\mathsf{T}}(\mathbf{z}_i - \mathbf{x}) dt$ . Now, to make the derivative error function emerge, one can add  $-\alpha_j \mathbf{x}^{\mathsf{T}} \nabla \phi_j(\mathbf{x}) + \mathbf{x}^{\mathsf{T}} \nabla f(\mathbf{x})$  to each side of the expressions. Together with the left-hand side of (B.2) this expression combines into  $\mathbf{e}_{\nabla f}(\mathbf{x})$ . In the right-hand side, these expressions can be included in the integrand since their value is constant with respect to the integration variable and considering that the integral is taken over the unit interval.

These manipulations allow to reorganize the equalities in (B.2) and rewrite them in the function of the operators  $\mathcal{Q}[f](\mathbf{x})$  and  $\mathcal{Q}_i[f](\mathbf{x})$ , which are defined as follows:

$$\begin{cases} \mathscr{Q}[f](\mathbf{x}) = \int_{0}^{1} \left( \nabla f \left( \mathbf{x} - t\mathbf{x} \right) - \nabla f \left( \mathbf{x} \right) \right)^{\mathsf{T}} (-\mathbf{x}) dt \\ \mathscr{Q}_{i}[f](\mathbf{x}) = \int_{0}^{1} \left( \nabla f \left( \mathbf{x} + t \left( \mathbf{z}_{i} - \mathbf{x} \right) \right) - \nabla f \left( \mathbf{x} \right) \right)^{\mathsf{T}} (\mathbf{z}_{i} - \mathbf{x}) dt \end{cases}$$
(B.3)

and in the function of the error functions,  $e_f(\mathbf{x})$  and  $\mathbf{e}_{\nabla f}(\mathbf{x})$ are given by the following:

$$\begin{cases} -\mathbf{x}^{\mathsf{T}} \mathbf{e}_{\nabla f} (\mathbf{x}) &= \mathcal{Q}[f] (\mathbf{x}) - \alpha_{k} \mathcal{Q}[\phi_{k}] (\mathbf{x}) - e_{f} (\mathbf{x}), \\ (\mathbf{z}_{i} - \mathbf{x})^{\mathsf{T}} \mathbf{e}_{\nabla f} (\mathbf{x}) &= \mathcal{Q}_{i}[f] (\mathbf{x}) - \alpha_{j} \mathcal{Q}_{i}[\phi_{j}] (\mathbf{x}) - e_{f} (\mathbf{x}), \end{cases}$$
(B.4)

of which the latter holds for all j > 0.

Introducing the operator,  $\delta Q_i[f](\mathbf{x}) = Q_i[f](\mathbf{x}) - Q[f](\mathbf{x})$ , and subtracting the first equation from the others yields the final expressions,

$$\mathbf{z}_{i}^{\top} \mathbf{e}_{\nabla f}(\mathbf{x}) = \delta \mathcal{Q}_{i}[f](\mathbf{x}) - \alpha_{k} \delta \mathcal{Q}_{i}[\phi_{k}](\mathbf{x}), \forall i > 0.$$
(B.5)

Considering that  $\mathcal{Z} \subset \mathcal{B}(0, \Delta)$ , it can be shown that the right-hand side is bounded by (see [40, 62]),

$$\begin{cases} \left| \mathcal{Q}[f](\mathbf{x}) - \alpha_{j} \mathcal{Q}[\phi_{j}](\mathbf{x}) \right| \leq \frac{1}{2} (\gamma_{f} + \gamma_{\phi}) \Delta^{2}, \\ \left| \mathcal{Q}_{i}[f](\mathbf{x}) - \alpha_{j} \mathcal{Q}_{i}[\phi_{j}](\mathbf{x}) \right| \leq 2 (\gamma_{f} + \gamma_{\phi}) \Delta^{2}, \\ \left| \delta \mathcal{Q}_{i}[f](\mathbf{x}) - \alpha_{j} \delta \mathcal{Q}_{i}[\phi_{j}](\mathbf{x}) \right| \leq \frac{5}{2} (\gamma_{f} + \gamma_{\phi}) \Delta^{2}, \end{cases}$$
(B.6)

where  $\gamma_f$  and  $\gamma_{\phi}$  are the Lipschitz constants of the functions  $\nabla f(\mathbf{x})$  and  $\alpha_k \nabla \phi_k(\mathbf{x})$ , given that  $\mathbf{x} \in \mathscr{B}(0, \Delta)$ .

The combination of the equations in (B.5) yields a bound on the derivative error function in the function of the interpolation points contained in  $\mathcal{Z}$ .

$$\begin{aligned} \left\| \begin{array}{c} \mathbf{z}_{1}^{\top} \\ \vdots \\ \mathbf{z}_{q}^{\top} \end{array} \right\| \left\| \mathbf{e}_{\nabla f} \left( \mathbf{x} \right) \right\|, \\ &= \left\| \mathbf{Z} \right\| \left\| \mathbf{e}_{\nabla f} \left( \mathbf{x} \right) \right\| \leq \frac{5}{2} \sqrt{q - 1\left( \gamma_{f} + \gamma_{\phi} \right)} \Delta^{2}. \end{aligned}$$
(B.7)

Since we already established a relation between both error functions in (B.5), the bound above allows one to determine a bound on  $e_f(\mathbf{x})$  as well, which is given by the following:

$$\left| \boldsymbol{e}_{f} \left( \mathbf{x} \right) \right| \leq \frac{1}{2} \left( \boldsymbol{\gamma}_{f} + \boldsymbol{\gamma}_{\phi} \right) \Delta^{2} + \frac{5}{2} \sqrt{q-1} \| \mathbf{Z} \|^{-1} \left( \boldsymbol{\gamma}_{f} + \boldsymbol{\gamma}_{\phi} \right) \Delta^{3}.$$
(B.8)

Now, upon introducing the scaled interpolation matrix  $\Delta \|\widehat{\mathbf{Z}}\| = \|\mathbf{Z}\|$ , we have established Taylor-like bounds on both the error value functions that are related to the inverse value of  $\|\widehat{\mathbf{Z}}\|$ . These bounds are, in fact, solely determined by the interpolation set geometry.

$$\begin{cases} \left\| \mathbf{e}_{\nabla f} \left( \mathbf{x} \right) \right\| \leq \frac{5}{2} \sqrt{q-1} \left\| \widehat{\mathbf{Z}} \right\|^{-1} \left( \gamma_{f} + \gamma_{\phi} \right) \Delta \\ \left\| \mathbf{e}_{f} \left( \mathbf{x} \right) \right\| \leq \left( \frac{5}{2} \sqrt{q-1} \left\| \widehat{\mathbf{Z}} \right\|^{-1} + \frac{1}{2} \right) \left( \gamma_{f} + \gamma_{\phi} \right) \Delta^{2} \end{cases}$$
(B.9)

#### C. Some Simplex-Related Defenitions

Consider a set of n + 1 points  $\mathcal{Z} = \{\mathbf{z}_0, \dots, \mathbf{z}_n\}$  associated with a set of coordinates in an *n* dimensional space. Assume that these points are linearly independent so that the column vectors  $\{\mathbf{z}_1 - \mathbf{z}_0, \dots, \mathbf{z}_n - \mathbf{z}_0\}$  are linearly independent. A simplex is, therefore, the *n*-dimensional generalization of a triangle. In simplex terminology, one has that each point from  $\mathcal{Z}$  determines a vertex, each couple of vertices determines an edge, and each subset of *n* vertices determines a face. Thus, an *n*-simplex has n + 1 vertices, n(n + 1)/2 edges, and n + 1 faces. Conveniently, it follows that each face of an n + 1-simplex constitutes an *n*-simplex itself.

According to the general convention, when speaking of the *n*-simplex that is determined by  $\mathcal{Z}$ , one refers to the convex set determined by  $\mathcal{S}$ .

$$\mathcal{S} = \left\{ \sum_{i=0}^{n} \theta_i \mathbf{z}_k | \sum_{i=0}^{n} \theta_i = 1 \land \theta_i \ge 0, \forall i \right\}.$$
(C.1)

Now, as it may be inconsistent with this general convention, in this work, we refer to the set of vertices  $\mathcal{Z}$  when speaking of the simplex. However, when we speak of the hypervolume of that simplex, we do, in fact, refer to the hypervolume of the convex set, S.

The volume *V* of the simplex determined by  $\mathcal{Z}$  is defined as the determinant of the associated column vector matrix  $[\mathcal{Z}]$  normalized by the factor *n*!.

$$V(\mathscr{Z}) = \frac{1}{n!} \det([\mathscr{Z}]), \qquad (C.2)$$

where the operator [.] is defined as  $[\mathcal{Z}] = [\mathbf{z}_1 - \mathbf{z}_0 \cdot \mathbf{z}_n - \mathbf{z}_0].$ 

In the following subsection, we mention some relevant simplex geometries and properties.

Orthocentric Simplex. A simplex for which the following holds are called orthocentric simplices: each edge is perpendicular to all the edges it does not meet [63]. Orthocentric simplices possess a unique orthocenter. In the special cases where  $n \le 2$ , every set of points that determines a simplex is orthocentric as well.

In [50], it is shown that an orthocentric simplex maximizes the volume satisfying a predetermined orthocenter, **o**, and distances from the simplex vertices to the orthocenter  $d_i = ||\mathbf{z}_i - \mathbf{o}||$ . These conditions determine the simplex,  $\mathcal{Z}$ , within an isometry. It is this very property that we exploit to maximize the volume of the union of an existing set and an expansion set generated by the procedure Simexpand, Algorithm 4. The expansion set is constructed exactly so that all of its edges are perpendicular to the edges determined by the existing set. Given the freedom of the edge lengths, albeit that the union of existing and expansion set is restricted to an *n*-sphere, these are determined so to maximize the volume.

Standard Simplex. A standard *n*-simplex is obtained when *n* of its edges are orthogonal and have the same length, *l*. A convenient coordinate representation of a standard *n*-simplex is given by the origin, 0, and the scaled coordinates of the *n* basis vectors,  $le_i$ . It is easily verified that the n(n-1)/2 nonorthogonal edges have length  $\sqrt{2l}$ .

We also note that the standard simplex is not an orthocentric simplex.

Regular Simplex. A regular *n*-simplex with side length l is obtained when all edges have the same length. From this definition, it follows immediately that any subset of m + 1 points from that simplex determines a regular *m*-simplex.

If we consider all simplices  $\mathcal{X}$  whose vertices are contained within an *n*-hypersphere with radius  $\Delta$ , it can be shown that if a simplex's volume is maximized, it is an element from the class of regular simplices [63]. Moreover, it can be shown that the regular simplex is an orthocentric simplex.

It is worth mentioning that a regular *n*-simplex is embedded in a standard n + 1-simplex as the face determined by the n(n + 1)/2 nonorthogonal edges of the standard n + 1-simplex. The edge length of a regular *n*-simplex that is embedded in a standard n + 1-simplex with an orthogonal edge length, *l*, is  $\sqrt{2l}$  in accordance with the nonorthogonal edge length of a standard simplex, with an edge length *l* as determined above.

Volume of the Regular *n*-Simplex. Let  $V_n^s(l)$  be the hypervolume occupied by the standard *n*-simplex with an orthogonal edge length *l*. Following the definition given in (C.4), it is easily verified that  $V_{n+1}^s(l) = 1/(n+1)!l^{n+1}$ . Now,

consider that the face determined by all the nonorthogonal edges of the standard n + 1-simplex constitutes a regular n-simplex. The distance from the intersection of the orthogonal edges, i.e., origin 0 according to the coordinate representation in C.2, to the face that determines the regular n-simplex is given by  $\sqrt{n + 1/n} + 1l$ . Since the hypervolume of a standard n-simplex can also be determined using the definition of the hyperpyramid, i.e., by multiplying the hypervolume of one of its faces with the corresponding altitude divided by n, the hypervolume of the regular n-simplex  $V_n^r(l)$  can be determined. Recall that the edge length of the regular n-simplex is  $\sqrt{2l}$  and that, therefore, the result must be scaled by a factor  $\sqrt{2^n}$ .

$$\frac{1}{n+1} \cdot \frac{\sqrt{n+1}}{n+1} l \cdot V_n^r(\sqrt{2} l) = V_{n+1}^s(l),$$
  
$$\Rightarrow V_n^r(l) = \frac{\sqrt{n+1}}{n!} \frac{l^n}{\sqrt{2^n}}.$$
 (C.3)

Radius of the Circumscribed n-Sphere. The radius,  $r_n(l)$ , of the *n*-sphere circumscribing a regular *n*-simplex with side length *l* can be determined reconsidering the geometric relations between a standard n + 1-simplex and a regular *n*-simplex described to determine the hypervolume. In a regular *n*-simplex, the distance between the unique orthocenter and a vertex is equal to the radius of the circumscribed *n*-sphere. It holds that the altitude of the face determined by the nonorthogonal edges and one of the orthogonal edges of the standard n+1-simplex constitute a right triangle with the orthocenter of the associated regular *n*-simplex and a common vertex. Note that again the edge length of this *n*-simplex is  $\sqrt{2l}$  and that, therefore, the result must be scaled with by a factor  $\sqrt{2}$ .

$$l^{2} = r_{n} (\sqrt{2} l)^{2} + \frac{1}{n+1} l^{2} \Rightarrow r_{n} (l) = \frac{\sqrt{n}}{\sqrt{n+1}} \frac{l}{\sqrt{2}}.$$
 (C.4)

Hence, the side length,  $l_n(\Delta)$ , of the regular *n*-simplex inscribed in an *n*-sphere with radius  $\Delta$  is given by  $l_n(\Delta) = \sqrt{2}\sqrt{n+1}/\sqrt{n}\Delta$ .

*Other Useful Relations*. From these results, we can derive some other equalities that are of interest for the procedure Simexpand described by Algorithm 4.

The radius of the circumscribed *m*-sphere of a regular *m*-simplex that is part of a regular *n*-simplex circumscribed by an *n*-sphere with radius  $\Delta$  is given by the following:

$$r_m^n(\Delta) = r_m(l_n(\Delta)) = \sqrt{\frac{m(n+1)}{(m+1)n}}\Delta.$$
 (C.5)

Hence, the distance,  $d_m^n(\Delta)$ , from the orthocenter of the regular *n*-simplex to a subset of m + 1 vertices determining such a regular *m*-simplex can be determined considering that,

$$\Delta^2 = d_m^n \left(\Delta\right)^2 + r_m^n \left(\Delta\right)^2 \Longrightarrow d_m^n \left(\Delta\right) = \sqrt{\frac{n-m}{(m+1)n}} \Delta.$$
(C.6)

This relation can be generalized. Consider, therefore, a regular *m*-simplex embedded in an *n*-dimensional space and inscribed in an *n*-sphere with radius  $\Delta$ . Its edge length must then be  $l_m(\Delta)$ . Now, if one would translate this *m*-simplex along any direction perpendicular to the *m*-dimensional subspace spanned by its edges over a distance  $\sigma\Delta$ , this *m*-simplex remains on the surface of the *n*-sphere as long as it is scaled by a factor  $\sqrt{1 - \sigma^2}\Delta$ .

## **Data Availability**

The numerical data used to support the findings of this study are available from the corresponding author upon request and are also available on the Githubpage associated to this article.

#### Disclosure

An earlier version of this research has been presented as a preprint.

#### **Conflicts of Interest**

The authors declare that they have no conflicts of interest.

#### Acknowledgments

The authors acknowledge the support of the FWO Research Project G.0D93.16N and the Flanders Make Project EMODO.

## References

- G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 7, p. 1995, 2019.
- [2] G. G. Wang, S. Deb, and L. D. S. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, no. 1, pp. 1–22, 2018.
- [3] G.-G. Wang, S. Deb, and L. S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence* (ISCBI), December 2015.
- [4] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [5] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [6] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [7] D. Karaboga, "Artificial bee colony algorithm," Scholarpedia, vol. 5, no. 3, p. 6915, 2010.
- [8] H. Rezaei, O. Bozorg-Haddad, and X. Chu, "Grey wolf optimization (gwo) algorithm," *Advanced Optimization by Nature-Inspired Algorithms*, Springer, Singapore, pp. 81–91, 2018.

- [9] J. Kennedy and E. Russell, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networksvol. volume 4, , pp. 1942–1948, IEEE, 1995.
- [10] N. Hansen, "The cma evolution strategy: a tutorial," 2016, https://arxiv.org/abs/1604.00772.
- [11] I. Loshchilov and F. Hutter, "Cma-es for hyperparameter optimization of deep neural networks," 2016, https://arxiv. org/abs/1604.07269.
- [12] A. Abdolmaleki, B. Price, N. Lau, L. Reis, and G. Neumann, "Deriving and improving cma-es with information geometric trust regions," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 657–664, ACM, Berlin, Germany, July 2017.
- [13] A. Abdolmaleki, R. Lioutikov, J. Peters, N. Lau, L. Reis, and G. Neumann, "Model-based relative entropy stochastic search," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3537–3545, Montreal, Quebec, Canada, January 2015.
- [14] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber, "Efficient natural evolution strategies," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 539–546, ACM, Montreal Québec Canada, July 2009.
- [15] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014.
- [16] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber, "Natural evolution strategies," in *Proceedings of the 2008 IEEE Congress* on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 3381–3387, IEEE, Hong Kong, China, June 2008.
- [17] T. Glasmachers, T. Schaul, y. Sun, D. Wierstra, and J. Schmidhuber, "Exponential natural evolution strategies," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 393–400, ACM, Portland Oregon USA, July 2010.
- [18] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, "Information-geometric optimization algorithms: a unifying picture via invariance principles," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 564–628, 2017.
- [19] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [20] M. W. Trosset and V. Torczon, Numerical Optimization Using Computer Experiments, Technical report, Institute for computer applications in science and engineering, Hampton, 1997.
- [21] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-free Optimization*, SIAM, philadelphia, 2009.
- [22] C. Audet and J. E. Dennis, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006.
- [23] S. . Le Digabel, "NOMAD: nonlinear optimization with the MADS algorithm," *Groupe d'études et de recherche en analyse des décisions*, vol. 37, no. 4, 2010.
- [24] Á. Bűrmen, J. Olenšek, and T. Tuma, "Mesh adaptive direct search with second directional derivative-based hessian update," *Computational Optimization and Applications*, vol. 62, no. 3, pp. 693–715, 2015.
- [25] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [26] G. Li, Q. Zhang, Q. Lin, and W. Gao, "A three-level radial basis function method for expensive optimization," *IEEE Transactions on Cybernetics*, 2021.

- [27] G. Li, Q. Zhang, J. Sun, and Z. Han, "Radial basis function assisted optimization method with batch infill sampling criterion for expensive optimization," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1664–1671, IEEE, Wellington, New Zealand, June 2019.
- [28] H. Dong, P. Wang, X. Yu, and B. Song, "Surrogate-assisted teaching-learning-based optimization for high-dimensional and computationally expensive problems," *Applied Soft Computing*, vol. 99, Article ID 106934, 2021.
- [29] H. Dong and Z. Dong, "Surrogate-assisted grey wolf optimization for high-dimensional, computationally expensive black-box problems," *Swarm and Evolutionary Computation*, vol. 57, Article ID 100713, 2020.
- [30] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, 2021.
- [31] S. M. Wild, R. G. Regis, and C. A. Shoemaker, "ORBIT: optimization by radial basis function interpolation in trustregions," *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3197–3219, 2008.
- [32] R. Oeuvray and M. Bierlaire, "Boosters: a derivative-free algorithm based on radial basis functions," *International Journal of Modelling and Simulation*, vol. 29, no. 1, pp. 26–36, 2009.
- [33] H.-M. Gutmann, "A radial basis function method for global optimization," *Journal of Global Optimization*, vol. 19, no. 3, pp. 201–227, Mar 2001.
- [34] M. J. D. Powell, "The NEWUOA Software for Unconstrained Optimization without Derivatives," in *Large-scale Nonlinear Optimization*, pp. 255–297, Springer, Berlin, Germany, 2006.
- [35] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, "A trust-region framework for managing the use of approximation models in optimization," *Structural Optimization*, vol. 15, no. 1, pp. 16–23, 1998.
- [36] H. B. Nielsen, S. N. Lophaven, and J. Sø ndergaard, A Matlab Kriging Toolbox. Informatics and Mathematical Modelling, Technical University of Denmark, Denmark, Europe, 2002.
- [37] M. J. D. Powell, "Recent research at cambridge on radial basis functions," in *Proceedings of the New Developments in Approximation Theory*, pp. 215–232, Springer, Germany, February 1998.
- [38] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, Statistical science, vol. 4, no. 4, pp. 409–423, 1989.
- [39] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 387–415, 2009.
- [40] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Geometry of interpolation sets in derivative free optimization," *Mathematical Programming*, vol. 111, no. 1, pp. 141–172, Jan 2008.
- [41] A. R. Conn, K. Scheinberg, and P. L. Toint, "On the convergence of derivative-free methods for unconstrained optimization," *Approximation theory and optimization: tributes to M. J. D. Powell*, pp. 83–108, Cambridge University Press, Cambridge, UK, 1997.
- [42] M. J. D. Powell, "On the Lagrange functions of quadratic models that are defined by interpolation," *Optimization Methods and Software*, vol. 16, no. 1-4, pp. 289–309, 2001.
- [43] O. Shir, J. Roslund, D. Whitley, and H. Rabitz, "Evolutionary hessian learning: forced optimal covariance adaptive learning (focal)," 2011, https://arxiv.org/abs/1112.4454.
- [44] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, Chennai, 2000.

- [45] J. E. Dennis and L. N. Vicente, "On the convergence theory of trust-region-based algorithms for equality-constrained optimization," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 927–950, 1997.
- [46] M. P. Uobyqa, "Unconstrained optimization by quadratic approximation," *Mathematical Programming*, vol. 92, no. 3, pp. 555–582, 2002.
- [47] A. R. Conn, K. Scheinberg, and P. L. Toint, "Recent progress in unconstrained nonlinear optimization without derivatives," *Mathematical Programming*, vol. 79, no. 1-3, p. 397, 1997.
- [48] M. J. D. Powell, Advances in Optimization and Numerical Analysis, Springer, Berlin, Germany, pp. 51–67, 1994.
- [49] D. J. J. Toal, Proper Orthogonal Decomposition & Kriging Strategies for Design, PhD thesis, University of Southampton, Southampton, England, 2009.
- [50] L. Gerber, "The orthocentric simplex as an extreme simplex," *Pacific Journal of Mathematics*, vol. 56, no. 1, pp. 97–111, 1975.
- [51] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A novel hybrid sequential design strategy for global surrogate modeling of computer experiments," *SIAM Journal on Scientific Computing*, vol. 33, no. 4, pp. 1948–1974, 2011.
- [52] R. Tanner, "Some content maximizing properties of the regular simplex," *Pacific Journal of Mathematics*, vol. 52, no. 2, pp. 611–616, 1974.
- [53] N. I. M. Gould, D. Orban, and P. L. Toint, "CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization," *Computational Optimization and Applications*, vol. 60, no. 3, pp. 545–557, 2015.
- [54] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software," ACM Transactions on Mathematical Software, vol. 7, no. 1, pp. 17–41, 1981.
- [55] S. Wild and R. Regis, "Bound-constrained orbit for matlab," https://http://www.mcs.anl.gov/wild/orbit/.
- [56] N. J. Higham, "The matrix computation toolbox," https:// www.ma.man.ac.uk/higham/mctoolbox.
- [57] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
- [58] J. J. Moré and S. M. Wild, "Benchmarking derivative-free optimization algorithms," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009.
- [59] MS Windows NT Kernel Description, https://github.ugent.be/ tjolefeb/IMORSE.gi.
- [60] J. D'Errico, Bound Constrained Optimization Using Fminsearch, https://nl.mathworks.com/matlabcentral/ fileexchange/8277-fminsearchbnd-fminsearchcon.
- [61] T. Lefebvre, F. De Belie, and G. Crevecoeur, "A radial basis function based optimization algorithm with regular simplex set geometry in ellipsoidal trust-regions," 2018, https://arxiv. org/abs/1805.11830.
- [62] J. E. Dennis, E. John, and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Chennai, 1996.
- [63] L. Gerber, "The altitudes of a simplex are associated," *Mathematics Magazine*, vol. 46, no. 3, pp. 155–157, 1973.