NOWPAC: A PROVABLY CONVERGENT DERIVATIVE-FREE NONLINEAR OPTIMIZER WITH PATH-AUGMENTED CONSTRAINTS

F. AUGUSTIN* AND Y. M. MARZOUK*

Abstract. This paper proposes the algorithm NOWPAC (Nonlinear Optimization With Path-Augmented Constraints) for nonlinear constrained derivative-free optimization. The algorithm uses a trust region framework based on fully linear models for the objective function and the constraints. A new constraint-handling scheme based on an inner boundary path allows for the computation of feasible trial steps using models for the constraints. We prove that the iterates computed by NOWPAC converge to a first-order critical point. We also discuss the convergence of NOWPAC in situations where evaluations of the objective function or the constraints are inexact, e.g., corrupted by numerical errors. We determine a rate of decay that the magnitude of these numerical errors must satisfy, while approaching the critical point, to guarantee convergence. In settings where adjusting the accuracy of the objective or constraint evaluations is not possible, as is often the case in practical applications, we introduce an error indicator to detect these regimes and prevent deterioration of the optimization results.

1. Introduction. In the design of industrial processes and other engineered systems, one often has to choose parameters $x \in \mathbb{R}^n$ in order to maximize performance while meeting prescribed requirements. The requirements and the performance objective may be available only as the result of black-box model evaluations, and the requirements may not be expressible in analytical form. To solve these problems, this paper introduces a new derivative-free approach for nonlinear constrained optimization. We generalize existing trust region methodologies by proposing a new scheme for handling general nonlinear constraints, and we prove convergence of the resulting algorithm to a first-order critical point. We also develop additional theory and an error indicator to account for inexact evaluations of the objective and constraints.

More precisely, we are interested in solving optimization programs of the form:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t.}\quad & c_i(x) \le 0,\ i = 1 \dots r \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function defining the quantity of interest and $c_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1 \dots r$, model the constraints imposed on the design parameters x. The constraints define the set of feasible points $X := \{x \in \mathbb{R}^n\ :\ c_i(x) \le 0,\ i = 1 \dots r\}$, consisting of all admissible designs. There exist many approaches for approximating the solutions of (1.1); see for instance [8, 9, 13, 17, 45]. The *constraints*, in particular, can be handled in various ways. One approach to enforcing the constraints is to replace the objective function by a merit function that penalizes the violation of the constraints [9, 13, 17]. The merit function can also be built using an inner barrier, penalizing proximity to the boundary of X and thus guaranteeing strict feasibility of the optimal design. In either case, good penalty parameters must be chosen to obtain an efficient algorithm. Current implementations use iterative approaches with increasing penalty parameters; see, for example, [44]. If the constraints are expensive to evaluate, reduced-order models can be used to reduce the computational costs [3, 25, 42]. Instead of using merit functions, the constraints can also be enforced via a Lagrange approach, which is often implemented in combination with sequential quadratic programming methods [9, 39]. Alternatively, [25, 26] introduce a filtering technique whose aim is to minimize the objective function and establish feasibility of the optimal point using a filter set of non-dominating pairs of objective values and constraint violations. There are also methods for nonlinear constrained optimization that do not rely on penalties or filters; for instance, [21, 27] introduce a trust-funnel method that sequentially reduces the value of the objective function and reduces constraint violations by taking steps tangential and normal to the feasible set.

Despite this wide array of constraint-handling approaches, most require derivative information on the objective function and the constraints, a situation which we do not wish to pursue in this

---

*Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA 02139, USA, {fmaugust,ymarz}@mit.edu.

paper. We will consider settings in which we have access to the objective function and the constraints only as black-box evaluations. Our setting is therefore derivative-free: we assume that derivatives of the objective and constraints are either unavailable or computationally too expensive to obtain. Moreover, we are interested in situations where we are not able to evaluate the objective function and the constraints exactly. Different methodologies have been proposed in this context, yet they typically assume increasing accuracy of the computations while approaching a critical point; see, e.g., [14, 15, 29]. In situations where the objective function and the constraints are only available as black-box evaluations, however, we may not be able to adjust or even bound the accuracy of these evaluations—i.e., the magnitude of the numerical errors or other perturbations to the functions f and c_i. This paper will therefore address the regime where we have neither control nor *a priori* knowledge on the inexactness of function evaluations.

The development of derivative-free optimization methods began in the 1960s, when Hooke and Jeeves [31] and Nelder and Mead [43], see also [61], were among the first to propose local direct search methods which only require black-box evaluations of the objective function. Since then, many other direct search methods have been proposed. For unconstrained programming we refer to [64, 65] and the references therein. Some direct search methods have been applied in the context of inexact function evaluations: for instance, the implicit filtering method [12, 35].

There are also derivative-free algorithms for computing the solutions of constrained nonlinear programs. Jones et al. [33] proposed the DIRECT method for derivative-free optimization with box constraints and studied its convergence. Later, this method and its convergence theory were extended to general nonlinear constraints; see [24]. Other direct search methods for constrained optimization problems have been proposed in [22, 40, 41]; these have been progressively generalized from local optimization with linear constraints, to local optimization with nonlinear constraints, and finally to global optimization. Another direct search method is the mesh adaptive direct search method (MADS) as introduced in [1]; in MADS the constraints are treated by an extreme barrier [4, 5] or by a progressive barrier [6] method.

A different class of derivative-free optimization methods approximate the objective function using local surrogate models. COBYLA [46, 47], for instance, is a widely used algorithm based on linear models of the objective. It handles constraints using a penalty approach based on a linear approximation. In [42], a low-fidelity surrogate model for the constraints is used to build a merit function based on a quadratic penalty. However, a rigorous proof that these methods converge to solutions of (1.1) is not available. Surrogate models based on radial basis functions have also been successfully applied in a derivative-free trust region setting [52, 53, 66, 67]. A recent effort [53] in this line of work discusses the derivative-free method COBRA for nonlinear constrained optimization, wherein a safety margin is added to the constraints. A prominent difference between COBRA and the present work lies in our adaptive handling of constraints via an inner boundary path, as opposed to the constant offset used in COBRA. A generalization of the trust-funnel method to derivative-free constrained optimization is presented in [57]. Here, a key difference with the present work lies in our feasibility requirement for every trial step; the trust-funnel approach, on the other hand, allows for constraint violations which are reduced while approaching a critical point. We refer to [20, 38, 54] for further overviews of derivative-free approaches.

The main contributions of this paper are threefold. First, we present the algorithm NOW-PAC (Nonlinear Optimization with Path-Augmented Constraints), which is based on a trust region framework. The algorithm introduces a new way of handling nonlinear black-box constraints using an *inner boundary path*, which is an offset function to the constraints whose purpose is to locally convexify the feasible domain. Besides this convexification, the inner boundary path guides the next trial step to become feasible, and therefore acceptable according to our proposed algorithm. Second, we develop a rigorous proof of convergence of the intermediate points computed by NOWPAC to a first-order critical point. Third, we analyze the behavior of our algorithm in the presence of inexact evaluations of the objective function and the constraints. Specifically, we show that the magnitude of the errors must respect a particular asymptotic decay rate in order to guarantee convergence.

Moreover, we provide an error indicator to detect corrupted evaluations in cases where the adjustment of the accuracy level is not possible. In the latter case we propose early termination of the optimization to avoid deterioration of the approximated optimal designs, and to save unnecessary evaluations of the objective function and constraints. Other than via early termination, our analysis does not attempt to reduce the impact of inexact evaluations, or to quantify the error in the approximated optimal designs that is due to inexact evaluations; we leave these developments to future work.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction to the trust region methodology; for more details, the reader is referred to [16, 19, 20, 49, 50]. Section 3 presents the algorithm NOWPAC. In Section 4, we prove the convergence of the intermediate points computed by NOWPAC to a first-order critical point. Thereafter, in Section 5, we discuss practical aspects of the implementation of our proposed algorithm. The proof of convergence is presented under the assumption of accurate evaluations of the objective function and the constraints. In practice, however, we may be faced with irreducible errors in the evaluations. Section 6 thus discusses the behavior of NOWPAC in the latter setting, deriving the asymptotic bounds and error indicator described above. In Section 7 we numerically demonstrate NOWPAC using two test examples, the Schittkowski benchmark set [58, 59], and a model of an industrial tar removal process used in converting biomass to liquid fuel. Concluding remarks and a sketch of future work are given in Section 8.

2. The trust region framework. In this section we introduce the derivative-free trust region framework used to approximate the solution of (1.1). Trust region methods start from an initial point x_0 and compute a series of intermediate points $\{x_k\}_{k \in \mathbb{N}_0}$ that converge to a critical point x^*. To help compute x_{k+1}, trust region methods build surrogates of the objective function f and the constraints $\{c_i\}_{i=1}^r$, denoted by $m_{x_k}^f$ and $\{m_{x_k}^{c_i}\}_{i=1}^r$ respectively, within a neighborhood of the current point x_k. The point x_{k+1} is then determined from the surrogates as a point that suitably reduces the objective function while staying within the neighborhood of x_k and satisfying the constraints. The neighborhood is called the trust region, $B(x_k, \rho_k) := \{x \in \mathbb{R}^n : \|x - x_k\| \le \rho_k\}$, with trust region radius ρ_k, $k \in \mathbb{N}_0$. Note that there are many possible choices of surrogate; among these, polynomial response surfaces [19, 48, 49, 51] are widely used. But other approximation methods can be employed as well; for example, radial basis functions are used to create the surrogates in [67]. The particular choice of surrogate models $m_{x_k}^f$ and $\{m_{x_k}^{c_i}\}_{i=1}^r$ for the objective function and the constraints is beyond the scope of this work, and we will not go into details on how to compute them. In general, any surrogates that are twice continuously differentiable and that satisfy

$$\left| f(x+s) - m_x^f(x+s) \right| \le \kappa_f \rho^2 \tag{2.1a}$$

$$\left| c_i(x+s) - m_x^{c_i}(x+s) \right| \le \kappa_c \rho^2 \tag{2.1b}$$

$$\left\| \nabla f(x+s) - \nabla m_x^f(x+s) \right\| \le \kappa_{df} \rho \tag{2.1c}$$

$$\left\| \nabla c_i(x+s) - \nabla m_x^{c_i}(x+s) \right\| \le \kappa_{dc} \rho \tag{2.1d}$$

with constants κ_f, κ_c, κ_{df}, $\kappa_{dc} > 0$, for all $x + s \in B(x, \rho)$, $i = 1 \ldots r$, are admissible. In our implementation of NOWPAC, we use quadratic minimum-Frobenius-norm surrogates; see [51]. Models satisfying (2.1) are called *fully linear* within the trust region $B(x, \rho)$. All of the surrogates previously mentioned satisfy these conditions, if certain geometry conditions on the sampling points of the model are satisfied; see [18, 51, 67]. With the surrogates denoted by $m_x^f(x+s)$ and $m_x^{c_i}(x+s)$, the corresponding gradients and Hessians at $s = 0$ are g_x^f, $g_x^{c_i}$ and H_x^f, $H_x^{c_i}$, respectively, for $i = 1 \ldots r$. Additionally, we assume the following:

3

Assumption 2.1. *For every constraint $i = 1 \ldots r$, there exists a bounding function*

$$b_{c_i} : \begin{cases} \mathbb{R}^n \times \mathbb{R}^n \times [0,1] & \to & \mathbb{R} \\ (s; x, \rho) & \mapsto & b_{c_i}(s; x, \rho) \end{cases}$$

such that $m_x^{c_i}(x+s) \leq c_i(x+s) + b_{c_i}(s; x, \rho)$ for all $s \in B(0, \rho)$ and $(x, \rho) \in X \times [0,1]$. The bounding function is continuous in (x, ρ) and satisfies $b_{c_i}(0, x, \rho) = 0$ as well as $b_{c_i} \leq \kappa_{\lambda_1} \rho$ for a constant $\kappa_{\lambda_1} > 0$ sufficiently large. At every $(x, \rho) \in X \times [0,1]$, we also assume that $b_{c_i}(\,\cdot\,; x, \rho)$ is continuously differentiable with Lipschitz continuous gradient in $B(0,1)$ and convex in $B(0, \rho)$.

Assumption 2.1 is satisfied by many surrogate models. In particular, we point to Lemma A.1, where we explicitly construct the bounding function b_c for quadratic minimum-Frobenius-norm models.

Before we state the trust region algorithm in the next section, we introduce a few general assumptions on the objective function and the constraints.

Assumption 2.2. *The objective function f and the constraints $\{c_i\}_{i=1}^r$ satisfy:*
(a) *$\mathcal{L} := X \cap \{x \in \mathbb{R}^n \,:\, f(x) \leq f(x_0)\}$ is compact,*
(b) *f and c_i are continuously differentiable on $\mathcal{L}$ and have Lipschitz continuous gradients,*
(c) *$\|\nabla c_i(x)\| \geq \kappa_{bdc}$ for all x on the boundary of X,*
(d) *at every critical point x^* of f in X the Abadie constraint qualification condition, see [7], holds,*
(e) *at any point on the boundary of the feasible domain X with more than one active constraint, any two normals to the active constraints enclose an angle strictly less than π.*

Assumption 2.3. *There exists a constant $\kappa_{bh} > 0$ such that $\|H_x^f\| \leq \kappa_{bh}$ and $\|H_x^{c_i}\| \leq \kappa_{bh}$ for all $i = 1 \ldots r$.*
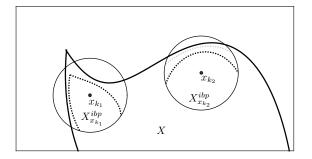
We note that Assumptions 2.2(a)–(d) ensure the existence of a solution of the optimization problem (1.1), which can be identified by first-order criticality conditions using linearized constraints. Assumption 2.2(e) excludes constraints that have the same tangent at a point on the boundary of the feasible domain, and hence excludes situations where two active constraints touch, as illustrated in Figure 3.1 (right). We note that Assumption 2.2(e) in particular excludes equality constraints. We further remark that, due to the continuity of ∇f and $\{\nabla c_i\}_{i=1}^r$ in Assumption 2.2, we also have $\|\nabla f\| \leq \kappa_{bdf}$ and $\|\nabla c_i\| \leq \kappa_{ubdc}$, $i = 1 \ldots r$, for all x in the compact set $\mathcal{L}$.

3. The algorithm NOWPAC. In this section we introduce the derivative-free algorithm NOWPAC for approximating local critical solutions of the nonlinear constrained problem (1.1). The notation and basic structure follow closely along the lines of [16, 20]; however we introduce significant changes in order to treat the nonlinear constraints as black-box evaluations. The outline of this section is as follows: In Section 3.1 we introduce some necessary notation and state assumptions on the sufficient descent of the objective model in every trust region step. In Section 3.2 we describe NOWPAC itself as Algorithm 1.

3.1. Preliminaries. First we introduce an offset to the constraint function, the *inner boundary path*,

$$h_x(x+d) : \begin{cases} \mathbb{R}^n & \to & \mathbb{R} \\ x+d & \mapsto & \varepsilon_b \, \|d\|^{\frac{2}{1+p}}, \end{cases} \tag{3.1}$$

with order reduction $p \in \,]0,1[$ and define the *inner-boundary-path-augmented local feasible domain* at $x \in X$ as

$$X_x^{ibp} := \{x + d \,:\, c(x+d) + h_x(x+d) \leq 0\} \cap B(x, 1). \tag{3.2}$$
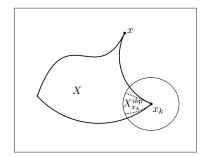
4

FIG. 3.1. *Left: Local convexification (black dotted lines) of the feasible domain X (solid line) around two designs x_{k_1} and x_{k_2}. Here an inner boundary path constant $\varepsilon_b = 10$ with $p = 0.2$ is chosen to overcome the negative curvature, i.e., the concavity, of the exact constraints c. The circle represents a ball around x_k with radius 1. The gray dotted line represents the local convexification resulting from a scaled inner boundary path constant $\varepsilon_b = 10\left(\frac{1}{2}\right)^2$, cf. (5.1). Right: Example of a feasible domain excluded by Assumption 2.2(e). Point x is not admissible since the interior of the associated inner-boundary-path-augmented feasible domain is empty due to the touching constraints.*

Note that the abbreviated notation c represents the corresponding expressions for all $\{c_i\}_{i=1}^r$, e.g., $c(x) \leq 0$ means $c_i(x) \leq 0$ for all $i = 1 \ldots r$. Subsequently, we will use this abbreviation also for the surrogate models, i.e., $m_k^c(x) + h_k(x) \leq 0$. We illustrate the inner-boundary-path-augmented local feasible domains X_x^{ibp} and the role of the inner boundary path h_x in Figure 3.1, where the inner boundary path constant $\varepsilon_b > 0$ is chosen large enough to achieve a local convexification of X_x^{ibp} (cf. Assumption 3.1 below). The importance of adding the inner boundary path h_x to m_x^c will become obvious in the discussion of the convergence of Algorithm 1 in Section 4. To quickly motivate the inner boundary path, however, we remark that it serves mainly two purposes: first, it locally convexifies the constraints around the point $x \in X$, as described in Assumption 3.1; second, it helps push all iterates away from the boundary and towards the inner part of the feasible domain X. We only use one common inner boundary path constant ε_b for all constraints to keep notation simple, but it is straightforward to extend the subsequent analysis to a set $\{\varepsilon_b^i\}_{i=1}^r$ of individual inner boundary path constants for each constraint c_i, $i = 1 \ldots r$.

Next we define the extended model

$$M_x^c(x) := \begin{cases} m_x^c(x) & \text{if } x \in B(x, \rho) \\ \bar{m}_x^c(x) & \text{if } x \in B(x, 1) \backslash B(x, \rho), \end{cases} \tag{3.3}$$

where $\bar{m}_x^c$ is a smooth extension of the local surrogates m_x^c to the ball $B(x, 1)$, if $\rho < 1$. If $\rho \geq 1$, then M_x^c is simply the fully linear surrogate model m_x^c within $B(x, \rho)$. Based on the extended surrogate model (3.3) we define the *approximated feasible domain*

$$X_x := \{x + d \ : \ M_x^c(x + d) + h_x(x + d) \leq 0\} \cap B(x, \max\{\rho, 1\}). \tag{3.4}$$

In order to have a reasonable approximation of X_x^{ibp} by the approximated feasible domain $X_x \cap B(x, 1)$ for vanishing trust region radii $\rho \to 0$, we assume that M_x^c is continuously differentiable with Lipschitz continuous gradient in $B(x, 1)$ with $|\bar{m}_x^c(x) - c(x)| \leq \kappa_{\lambda_2} \rho_k$ for a sufficiently large constant $\kappa_{\lambda_2} > 0$. Finally we define the *inner and outer approximation sets*

$$X_{(x,\rho)}^+ := \{x + d \ : \ c(x + d) + h_x(x + d) + b_c(d; x, \rho) \leq 0\} \cap B(x, 1), \tag{3.5}$$

$$X_{(x,\rho)}^- := \{x + d \ : \ c(x + d) + h_x(x + d) - \kappa_{\lambda_3}\rho \qquad \leq 0\} \cap B(x, 1),$$

where $\kappa_{\lambda_3} > 0$ is chosen large enough such that $X_{(x,\rho)}^+ \subseteq X_x \subseteq X_{(x,\rho)}^-$, justifying the labels of 'inner' and 'outer' approximation. We will make use of this inclusion of the sets $X_{(x,\rho)}^+$, X_x, $X_{(x,\rho)}^-$ in

5

symbol	range	description
ε_b	$]0, \infty[$	inner boundary path constant$^{(\star)}$
η_0	$[0, 1[$	step rejection parameter
η_1	$[\eta_0, 1[$	step acceptance parameter$^{(\star)}$
γ_{inc}	$]1, \infty[$	increment factor for trust region
γ	$]0, 1[$	decrease factor for trust region
ε_c	$]0, \infty[$	lower bound on trusted criticality measure
ω	$]0, 1[$	decrease factor for trust region
μ	$]0, \infty[$	factor for bound on trust region radius by α

Lemma 4.2. Note that the inner and outer approximations $X^+_{(x,\rho)}$ and $X^-_{(x,\rho)}$ to the approximated feasible domain X_x are defined by lower and upper bounds on *all* possible local surrogate models and are therefore independent of the particular surrogate model that defines X_x. Henceforth we will use a single subscript k for any of the quantities introduced above when referring to a particular point $x_k \in X$, i.e., $X^{ibp}_k := X^{ibp}_{x_k}$, $X_k := X_{x_k}$, as well as $m^f_k := m^f_{x_k}$ and $m^c_k := m^c_{x_k}$, etc.

ASSUMPTION 3.1. *Assume that $\varepsilon_b > 0$ is large enough such that the sets X^{ibp}_x and X_k and the inner and outer approximations $X^\pm_{(x,\rho)}$ are strictly convex.*

Lemma A.2 shows the existence of an inner boundary path constant ε_b such that Assumption 3.1 is satisfied.

Before we state the algorithm we have to define a measure for criticality, i.e., a measure for the proximity of the current iterate x_k to a critical point:

$$\alpha_k(\rho_k) := \frac{1}{\rho_k} \left| \min_{\substack{x_k + d \in X_k \\ \|d\| \leq \rho_k}} \left\langle g^f_k, d \right\rangle \right|. \tag{3.6}$$

Since α_k uses models for both the objective and the constraints, we call it the *approximated criticality measure*. We refer to Section 4.1 for a detailed discussion of α_k. Besides a measure for criticality, we also need to specify an initial point x_0 and a maximal trust region radius $\rho_{\max} > 0$. Moreover, we set the initial trust region radius to $\rho_0 \in]\rho_{\min}, \rho_{\max}]$, $\rho_{\min} > 0$, and specify technical parameters as shown in Table 3.1. As already mentioned in Section 2, our trust region algorithm computes, starting from the current iterate x_k, a trial step $x_k + s_k \in X_k \cap B(x_k, \rho_k)$. For the computation of the trial step s_k we impose the following assumptions:

ASSUMPTION 3.2. *The trial step s_k computed by Algorithm 1 satisfies*
(a) $m^f_k(x_k) - m^f_k(x_k + s_k) \geq \mu_1 \alpha_k(\rho_k)\rho_k$, and
(b) $\|s_k\| \geq \min\left\{\mu_2\rho_k^{1+q}, \mu_3\right\}$
for $q < p$, $\mu_1, \mu_2, \mu_3 \in]0, 1]$, as well as the feasibility and trust region condition $x_k + s_k \in X_k \cap B(x_k, \rho_k)$.

The first assumption ensures that the step yields a sufficient descent in the objective model, whereas the second assumption keeps the step sizes from becoming too small. Note that the parameter p in Assumption 3.2 is the same as in the definition of the inner boundary path (3.1). We defer discussion of the existence of a step s_k satisfying Assumptions 3.2 to Section 5.

3.2. The algorithm. We present the complete derivative-free trust region algorithm NOW-PAC in Algorithm 1. We call iteration k *successful* if the acceptance ratio r_k exceeds the threshold

---
Algorithm 1: Nonlinear Optimization With Path-Augmented Constraints
---

1 Construct the initial fully linear models $m_0^f(x_0 + s)$, $m_0^c(x_0 + s)$.

2 for *k = 0, 1, . . .* **do**

/* STEP 1: Criticality step */

3 **if** $\alpha_k(\rho_k) \leq \varepsilon_c$ **then**

4   **if** *m_k^f and m_k^c are not fully linear in $B(x_k, \rho_k)$ or $\rho_k > \mu\alpha_k(\rho_k)^{\frac{1}{q}}$* **then**

5   Set $\rho_k = \omega\rho_k$

6   Construct fully linear models m_k^f and m_k^c

7   Go to line 4

8   **end**

9 **end**

/* STEP 2: Step calculation */

10 Compute a trial step s_k that satisfies Assumptions 3.2

/* STEP 3: Check feasibility of trial point */

11 **if** $c(x_k + s_k) > 0$ **then**

12   Set $\rho_k = \gamma\rho_k$ and update m_k^f and m_k^c accordingly to obtain fully linear models

13   Go to STEP 1

14 **end**

/* STEP 4: Check acceptance of trial point */

15 Compute $r_k = \frac{f(x_k) - f(x_k + s_k)}{m_k^f(x_k) - m_k^f(x_k + s_k)}$

16 **if** $r_k \geq \eta_0$ **then**

17   Set $x_{k+1} = x_k + s_k$

18   Include x_{k+1} into the node set and update the models to m_{k+1}^f and m_{k+1}^c

19 **else**

20   Set $x_{k+1} = x_k$, $m_{k+1}^f = m_k^f$ and $m_{k+1}^c = m_k^c$

21 **end**

/* STEP 5: Trust region update */

22 Set $\rho_{k+1} = \begin{cases} \gamma_{inc}\rho_k & \text{if} \quad r_k \geq \eta_1 \\ \rho_k & \text{if} \quad \eta_0 \leq r_k < \eta_1, \\ \gamma\rho_k & \text{if} \quad r_k < \eta_0. \end{cases}$

/* STEP 6: Model improvement */

23 **if** $r_k < \eta_0$ **then**

24   Improve the quality of the models m_{k+1}^f and m_{k+1}^c

25 **end**

26 end

η_1, whereas we call it *acceptable* if $\eta_0 \leq r_k < \eta_1$. Note that we do not specify a stopping criterion to terminate the algorithm. The usual approach in derivative-free trust region algorithms is to stop whenever the trust region radius falls below a prescribed threshold $\rho_{\min} > 0$. We will see in Section 4.3 that this is a reasonable stopping criterion for NOWPAC as well, since $\rho_k \to 0$ for the sequence $\{x_k\}_k$ converging to a critical point x^*. We therefore insert the line

> **22a** **if** $\rho_{k+1} < \rho_{min}$ **then** *stop*.

into Algorithm 1 below for the actual implementation of NOWPAC. However, since we want to

examine the asymptotic behavior of the iterates as $k \to \infty$, we do not include a stopping criterion in the forthcoming theoretical investigations. For more practical aspects of the implementation of NOWPAC see Section 5.

Finally we note that the construction of fully linear models in lines 6, 18, and 24 of Algorithm 1 can be completed in a finite number of steps. In particular, the model improvement STEP 6 computes at most finitely many intermediate points before the models become fully linear; see [20].

4. Convergence to first-order critical points. In this section we prove convergence of the intermediate points $\{x_k\}_k$ generated by Algorithm 1 to a first-order critical point x^* of (1.1). We subdivide this section into three parts. First, in Section 4.1, we show that if the approximated criticality measure $\alpha_k(\rho_k)$ evaluated at intermediate points x_k vanishes to zero, then $x^* = \lim_{k \to \infty} x_k$ is a first-order critical point. For convergence, we then have to prove that Algorithm 1 indeed generates a sequence of intermediate points on which the approximated criticality measure converges to zero. In Section 4.2, we show that Algorithm 1 computes $\{x_k\}_k$ without getting trapped in an infinite loop of infeasible or rejected steps in STEP 3 and STEP 4. Thereafter, in Section 4.3, we complete the proof of convergence by showing that the sequence of intermediate points generated by Algorithm 1 converges to a first-order critical point. The general ideas within Section 4 follow along the lines of [16]. But we develop additional arguments in order to show convergence in the case of approximated constraint functions.

4.1. The criticality measure. When an optimal point x^* is located at the boundary of the feasible set X, it is well known that the gradient is not necessarily an appropriate indicator for criticality. We therefore rely on the fact that $x^* \in X$ is a critical point if and only if

$$- \nabla f(x^*) \in N(x^*), \tag{4.1}$$

where $N(x) := \{y \in \mathbb{R}^n \ : \ \langle y, u - x \rangle \leq 0, \ \forall u \in X\}$ denotes the normal cone of the set of feasible points X at point x; due to Assumption 2.2(d) we can use the linearized constraints at x^* to characterize the normal cone. Moreover, note that $N(x) = \{0\}$ whenever x is an inner point of X and (4.1) reduces to the gradient criterion $\|\nabla f(x^*)\| = 0$ for local first-order optimality. We now define the exact criticality measure,

$$\mathfrak{A}[x] := \left| \min_{x+d \in X_x^{ibp}} \langle \nabla f(x), d \rangle \right|, \tag{4.2}$$

which gives the maximal possible decrease of the linearized objective function within X_x^{ibp}, i.e., the inner-boundary-path-augmented local feasible domain (3.2). Note that the criticality subproblem (4.2) is linear in d with $d = 0$ always a feasible point, since $x \in X_x^{ibp}$ by definition. Thus the optimal value of the criticality subproblem, $\min_{x+d \in X_x^{ibp}} \langle \nabla f(x), d \rangle$, is necessarily less than or equal to zero and we can write

$$\mathfrak{A}[x] = - \langle \nabla f(x), d^* \rangle$$

for $d^* := \arg \min_{x+d \in X_x^{ibp}} \langle \nabla f(x), d \rangle$. We will frequently use this relation between the absolute value and the negation of the optimal value of the criticality subproblem within the subsequent proofs.

The following lemma shows that $\mathfrak{A}[x] = 0$ is a reasonable indicator for criticality of the point $x \in X$.

LEMMA 4.1. *Under Assumptions 2.2, the point $x \in X$ is critical if and only if $\mathfrak{A}[x] = 0$.*

Proof. First note that due to Assumptions 2.2(c, e) and the inner boundary path $h_x(\cdot)$ vanishing superlinearly (with exponent $\frac{2}{1+p}$) around the center point, the interior of X_x^{ibp} is always non-empty, as illustrated in Figure 3.1 (right). Thus, because of the convexity of X_x^{ibp}, the Slater condition holds for X_x^{ibp}. Now observe that the gradient of $h_x(x)$ is zero, which means that the normal cones

8

to X and X_x^{ibp} at x are identical. The criticality conditions for the criticality subproblem in (4.2) can be expressed as

$$0 \in 2\lambda d + \nabla f(x) + N(x+d), \quad x+d \in X_x^{ibp},$$

$$\|d\| \leq 1 \quad \text{and} \quad \lambda(d^T d - 1) = 0. \tag{4.3}$$

We see that if $\mathfrak{A}[x] = 0$, then $d = 0$ is an optimal solution to the criticality subproblem in (4.2). Inserting $d = 0$ into (4.3) yields (4.1), i.e., $-\nabla f(x) \in N(x)$. On the other hand, if (4.1) holds, then the above conditions (4.3) are satisfied with $d = 0$ and $\lambda = 0$, implying $\mathfrak{A}[x] = 0$. □

At this point the exact criticality measure $\mathfrak{A}[x_k]$ depends on the gradient ∇f of the objective function and the exact constraint functions c. In the context of derivative-free optimization we know neither the gradient nor the exact algebraic structure of the constraint functions. Thus, we are not able to evaluate the exact criticality measure and must modify it in order to obtain a criticality measure that we can evaluate. Accordingly, we replace X_k^{ibp} with X_k and substitute the model gradient g_k^f for the exact gradient $\nabla f(x_k)$. The result is the approximated criticality measure $\alpha_k(\rho_k)$ given in (3.6) and used in Algorithm 1. At this point it remains to show that the approximated criticality measure serves our need to drive iterates of the algorithm to a critical point of (1.1). We address this problem with the following lemma.

LEMMA 4.2 (Relation between the exact and approximated criticality measures). *Under Assumption 2.2, let $\{x_k\}_k \in X$ be a sequence of points in the feasible domain X and $\{\rho_k\}_k$ a sequence of trust region radii with $\lim_{k\to\infty} \rho_k = 0$. It holds that*

$$\lim_{k\to\infty} \alpha_k(\rho_k) = 0 \quad \Rightarrow \quad \lim_{k\to\infty} \mathfrak{A}[x_k] = 0.$$

Proof. Due to $\lim_{k\to\infty} \alpha_k(\rho_k) = 0$ there exists an index $\hat{k}$ such that $\alpha_k(\rho_k) \leq \varepsilon_c$ for all $k \geq \hat{k}$. In this case STEP 1 of Algorithm 1 ensures that the models m_k^f and m_k^c are fully linear on $B(x_k, \rho_k)$ with $\rho_k \leq \mu \alpha_k(\rho_k)^{\frac{1}{p}}$ for all $k \geq \hat{k}$, in particular yielding $\lim_{k\to\infty} \rho_k = 0$. Since we are only interested in the asymptotic behavior of the criticality measures, we assume without loss of generality that $\rho_k < 1$.

First, define the intermediate criticality measures

$$\mathfrak{A}_1[x_k] := \left| \min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \langle \nabla f(x_k), s \rangle \right| \quad \text{and} \quad \mathfrak{A}_2[x_k] := \left| \min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \langle g_k^f, s \rangle \right|.$$

Note that the difference between $\mathfrak{A}_2[x_k]$ and $\rho_k \alpha_k(\rho_k)$ is that the former is constrained by X_k rather than by $X_k \cap B(x_k, \rho_k)$. The difference between $\mathfrak{A}_2[x_k]$ and $\mathfrak{A}_1[x_k]$ is in the gradient of the criticality subproblem, and the difference between $\mathfrak{A}_1[x_k]$ and $\mathfrak{A}[x_k]$ lies in the introduction of the approximated feasible domain X_k. In order to prove the assertion of the lemma we use the triangle inequality,

$$\mathfrak{A}[x_k] \leq |\mathfrak{A}[x_k] - \mathfrak{A}_1[x_k]| + |\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k]| + \mathfrak{A}_2[x_k],$$

and show that each term on the right-hand side vanishes for decreasing trust region radius ρ_k and for a vanishing approximated criticality measure $\alpha_k(\rho_k)$. For this we consider the combined sequence $\{(x_k, \rho_k)\}_k$ of intermediate points and trust region radii as computed by Algorithm 1. Furthermore we define

$$\mathfrak{A}_1^{\pm}[x, \rho] := \left| \min_{x+s\in X_{(x,\rho)}^{\pm}} \langle \nabla f(x), s \rangle \right|,$$
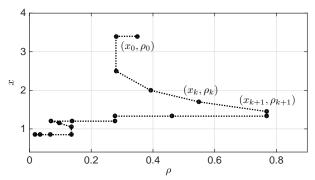
FIG. 4.1. *Typical path in the domain $X \times [0, \rho_{max}]$ as computed by Algorithm 1.*

which, according to Lemma A.3, are continuous functions in (x, ρ). Lemma A.3 establishes the continuity of $\mathfrak{A}_1^{\pm}$ on the whole domain $X \times [0, \rho_{max}]$, which then holds in particular for the path through $\{(x_k, \rho_k)\}_k$ as computed by Algorithm 1, as depicted in Figure 4.1. Note that $X_{(x_k,0)}^+ = X_{(x_k,0)}^- = X_k^{ibp}$ for all $x_k \in X$, i.e., $\mathfrak{A}_1^+[x_k, \rho_k] = \mathfrak{A}_1^-[x_k, \rho_k] = \mathfrak{A}[x_k]$ for $\rho_k = 0$. It then follows from the continuity of $\mathfrak{A}_1^+$ and $\mathfrak{A}_1^-$ that for every $\varepsilon > 0$ there exists a k_ε such that

$$|\mathfrak{A}_1^+[x_k, \rho_k] - \mathfrak{A}[x_k]| \leq \varepsilon \quad \text{and} \quad |\mathfrak{A}_1^-[x_k, \rho_k] - \mathfrak{A}[x_k]| \leq \varepsilon$$

for all $k \geq k_\varepsilon$. Due to $X_{(x_k,\rho_k)}^+ \subseteq X_k \subseteq X_{(x_k,\rho_k)}^-$ we have[1]

$$\mathfrak{A}_1^+[x_k, \rho_k] \leq \mathfrak{A}_1[x_k] \leq \mathfrak{A}_1^-[x_k, \rho_k].$$

Subtracting $\mathfrak{A}[x_k]$ from this inequality yields

$$-\varepsilon \leq \mathfrak{A}_1^+[x_k, \rho_k] - \mathfrak{A}[x_k] \leq \mathfrak{A}_1[x_k] - \mathfrak{A}[x_k] \leq \mathfrak{A}_1^-[x_k, \rho_k] - \mathfrak{A}[x_k] \leq \varepsilon,$$

and thus $|\mathfrak{A}_1[x_k] - \mathfrak{A}[x_k]| \leq \varepsilon$ for $k \geq k_\varepsilon$.

Next we derive an upper bound on the difference between the two intermediate criticality measures $\mathfrak{A}_1[x_k]$ and $\mathfrak{A}_2[x_k]$. Since for $k \geq \hat{k}$ the model m_k^f is fully linear within the trust region $B(x_k, \rho_k)$ and we have that $\|\nabla f(x) - \nabla m_k^f(x)\| \leq \kappa_{df}\rho_k$ for all $x \in B(x_k, \rho_k)$, it holds in particular that $\|\nabla f(x_k) - g_k^f\| \leq \kappa_{df}\rho_k$. We now follow along the lines of [16, Lem. 3.5] to show that $|\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k]| \leq \kappa_{df}\rho_k$. Denote the solutions of the first and second intermediate criticality subproblems, $\mathfrak{A}_1[x_k]$ and $\mathfrak{A}_2[x_k]$, by

$$s_k := \underset{x_k+s \in X_k, \|s\| \leq 1}{\arg\min} \langle \nabla f(x_k), s \rangle \quad \text{and} \quad \hat{s}_k := \underset{x_k+s \in X_k, \|s\| \leq 1}{\arg\min} \left\langle g_k^f, s \right\rangle.$$

---

[1]It holds that $\min_{x_k+s \in X_{(x_k,\rho_k)}^+} \langle \nabla f(x_k), s \rangle \geq \min_{x_k+s \in X_k, \|s\| \leq 1} \langle \nabla f(x_k), s \rangle \geq \min_{x_k+s \in X_{(x_k,\rho_k)}^-} \langle \nabla f(x_k), s \rangle.$
Multiplication with -1 yields $\mathfrak{A}_1^+[x_k, \rho_k] \leq \mathfrak{A}_1[x_k, \rho_k] \leq \mathfrak{A}_1^-[x_k, \rho_k].$

10

Let us first assume the case $\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k] \geq 0$. It follows that

$$
\begin{aligned}
0 \leq \mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k] &= \left\langle g_k^f, \hat{s}_k \right\rangle - \left\langle \nabla f(x_k), s_k \right\rangle \\
&= \left\langle g_k^f, \hat{s}_k \right\rangle - \left\langle g_k^f, s_k \right\rangle + \left\langle g_k^f - \nabla f(x_k), s_k \right\rangle \\
&\leq \left\langle g_k^f, \hat{s}_k \right\rangle - \left\langle g_k^f, s_k \right\rangle + \left\| g_k^f - \nabla f(x_k) \right\| \|s_k\| \\
&\leq \left\langle g_k^f, \hat{s}_k \right\rangle - \left\langle g_k^f, s_k \right\rangle + \kappa_{df}\rho_k,
\end{aligned}
$$

where we used the Cauchy-Schwarz inequality, the full linearity property (2.1c), and the fact that $\|s_k\| \leq 1$. Noting that $\langle g_k^f, \hat{s}_k \rangle \leq \langle g_k^f, s_k \rangle$ (since $\hat{s}_k$ is a minimum of the intermediate criticality subproblem associated with $\mathfrak{A}_2[x_k]$) yields the bound $\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k] \leq \kappa_{df}\rho_k$. The upper bound in the case where $\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k] \leq 0$ can be shown analogously by replacing the first line in the above inequality chain with $0 \leq \mathfrak{A}_2[x_k] - \mathfrak{A}_1[x_k] = \langle \nabla f(x_k), s_k \rangle - \langle \nabla f(x_k), \hat{s}_k \rangle + \langle \nabla f(x_k) - g_k^f, \hat{s}_k \rangle$.

In order to complete the proof we refer to Lemma A.4 where we show that $\mathfrak{A}_2[x_k] \leq \alpha_k(\rho_k)$. Then the assertion of the lemma follows from the assumption that $\lim_{k \to \infty} \alpha_k(\rho_k) = 0$. ☐

4.2. Successful iterations. From the definition of Algorithm 1 we see that all intermediate points x_k that are either not feasible or not acceptable are discharged within STEP 3 or STEP 4. In this section, we show that Algorithm 1 does not get trapped in an infinite loop of discharged iterations that result in premature convergence to a potentially non-critical point. First we show that STEP 3 in Algorithm 1 determines a feasible point if the trust region radius is sufficiently small. Then, we examine the second hurdle for a successful iteration: the step acceptance condition in STEP 4.

LEMMA 4.3. *Let m_k^c be fully linear on $B(x_k, \rho_k)$. STEP 3 in Algorithm 1 yields a feasible trial step if*

$$
\rho_k \leq \left(\frac{\varepsilon_b}{\kappa_c} \mu_2^{\frac{2}{1+p}} \right)^{\frac{1+p}{2(p-q)}}.
$$

Proof. Since $x_k + s_k \in X_k \cap B(x_k, \rho_k)$ we know that $m_k^c(x_k + s_k) + h_k(x_k + s_k) \leq 0$, i.e.,

$$
m_k^c(x_k + s_k) \leq -h_k(x_k + s_k) = -\varepsilon_b \|s_k\|^{\frac{2}{1+p}} \leq -\varepsilon_b \mu_2^{\frac{2}{1+p}} \rho_k^{2\frac{1+q}{1+p}}, \tag{4.4}
$$

where we used Assumption 3.2(b) in the last inequality. Moreover, from (4.4) and the fully linear property of m_k^c, we get

$$
c(x_k + s_k) \leq m_k^c(x_k + s_k) + \kappa_c \rho_k^2 \leq -\varepsilon_b \mu_2^{\frac{2}{1+p}} \rho_k^{2\frac{1+q}{1+p}} + \kappa_c \rho_k^2.
$$

Thus STEP 3 in Algorithm 1 yields a feasible trial step, i.e., $c(x_k + s_k) \leq 0$, if

$$
-\varepsilon_b \mu_2^{\frac{2}{1+p}} \rho_k^{2\frac{1+q}{1+p}} + \kappa_c \rho_k^2 \leq 0.
$$

Solving for ρ_k yields the assertion of the lemma. ☐

The next lemma shows that if the trust region radius falls below a certain threshold (given by the criticality measure and the threshold in Lemma 4.3), then the trial step will be accepted. In other words, if the current design is not a critical point, then we can always find a successful trial step.

11

LEMMA 4.4. *If m_k^f and m_k^c are fully linear on $B(x_k, \rho_k)$ and $\rho_k \leq A_k(\alpha_k)$, for*

$$A_k(\alpha_k) := \min\left\{ \frac{\mu_1 \alpha_k(\rho_k)(1 - \eta_0)}{2\kappa_f}, \; \left(\frac{\varepsilon_b}{\kappa_c} \mu_2^{\frac{2}{1+p}} \right)^{\frac{1+p}{2(p-q)}} \right\},$$

then iteration k will be acceptable or successful.

Proof. Since the assumptions of Lemma 4.3 are satisfied, Algorithm 1 passes STEP 3 with a feasible trial step. Moreover, from Assumption 3.2(a) we know that

$$m_k^f(x_k) - m_k^f(x_k + s_k) \geq \mu_1 \alpha_k(\rho_k)\rho_k.$$

Using the fully linear properties of the model m_k^f on $B(x_k, \rho_k)$ we get

$$
\begin{aligned}
|r_k - 1| &\leq \left| \frac{f(x_k + s_k) - m_k^f(x_k + s_k)}{m_k^f(x_k) - m_k^f(x_k + s_k)} \right| + \left| \frac{f(x_k) - m_k^f(x_k)}{m_k^f(x_k) - m_k^f(x_k + s_k)} \right| \\
&\leq \frac{2\kappa_f \rho_k^2}{\mu_1 \alpha_k(\rho_k)\rho_k} \leq 1 - \eta_0.
\end{aligned}
$$

Therefore $r_k \geq \eta_0$ and iteration k is acceptable or successful. $\square$

4.3. Proof of convergence. Having ensured that Algorithm 1 always finds an acceptable or successful feasible trial step, we now show convergence of the intermediate points $\{x_k\}_k$ to a first-order critical point x^*. Following the ideas in [16, 20] we establish a relation between the trust region radii $\{\rho_k\}_k$ and the criticality measures $\{\mathfrak{A}[x_k]\}_k$; we reason that $\lim_{k \to \infty} \rho_k = 0$, from which we eventually conclude $\lim_{k \to \infty} \mathfrak{A}[x_k] = 0$. We start by proving the technical auxiliary Lemma 4.5 where we show that if the approximated criticality measures $\{\alpha_k\}_k$ are bounded from below by a positive constant, then the sequence $\{\rho_k\}_k$ of trust region radii will also be bounded from below by a positive constant, cf. [20, Lem. 10.7].

LEMMA 4.5. *Suppose that there exists a constant $\kappa_1 > 0$ such that $\alpha_k(\rho_k) \geq \kappa_1$ for all k. Then there exists a constant $\kappa_2 > 0$ such that $\rho_k \geq \kappa_2$ for all k.*

Proof. By Lemma 4.4 (note that STEP 1, STEP 3, and STEP 6 in Algorithm 1 ensure full linearity of the models m_k^f and m_k^c after every reduction of the trust region radius) it holds that whenever ρ_k falls below the value

$$\bar{\kappa}_2 = A_k(\kappa_1), \tag{4.5}$$

the kth iteration is either acceptable or successful, and hence it holds that $\rho_{k+1} \geq \rho_k$. We conclude from (4.5) and the rules of STEPS 1, 3, and 5 that $\rho_k \geq \min\{\mu\omega\kappa_1^{\frac{1}{q}}, \gamma\bar{\kappa}_2\} =: \kappa_2$. $\square$

For notational convenience we denote the set of indices of all acceptable or successful steps by $\mathcal{S}$. In the next lemma we show convergence of Algorithm 1 to a first-order critical point if $|\mathcal{S}| < \infty$ (i.e., if there are only finitely many acceptable or successful steps).

LEMMA 4.6. *If $|\mathcal{S}| < \infty$, then $\lim_{k \to \infty} \mathfrak{A}[x_k] = 0$.*

Proof. First we show that $\lim_{k \to \infty} \rho_k = 0$ if $|\mathcal{S}| < \infty$. For this we note that STEP 6 in Algorithm 1 ensures full linearity of the models for the objective function and the constraints within every iteration after the last acceptable or successful iteration. Therefore, after the last acceptable or successful step, Algorithm 1 never increases the trust region radius ρ_k but reduces it either by a factor of some power of ω in STEP 1 or by a factor of γ in STEP 3 or STEP 5. It follows that $\lim_{k \to \infty} \rho_k = 0$. This in turn implies that $\lim_{k \to \infty} \alpha_k(\rho_k) = 0$; if the approximate criticality measures were bounded away from zero, then Lemma 4.4, for small ρ_{k+1}, guarantees that step $k + 1$ is either acceptable

12

or successful, yielding a contradiction to $|\mathcal{S}| < \infty$. The assertion of this lemma now follows from Lemma 4.2. $\square$

Thus far we have proved the convergence of Algorithm 1 in the case of $|\mathcal{S}| < \infty$. In the remainder of this section we extend the proof of convergence to $\mathcal{S}$ being a countably infinite set. To this end, we first show that the sequence of trust region radii $\{\rho_k\}$ converges to zero even if $\mathcal{S}$ is infinite in Lemma 4.7. This immediately implies the existence of a subsequence of approximated criticality measures $\{\alpha_k(\rho_k)\}_k$ that converges to zero. Finally we combine all results to prove the convergence of Algorithm 1 towards a first-order critical point in Theorem 4.8.

LEMMA 4.7. *It holds that* $\lim_{k\to\infty} \rho_k = 0$.

Proof. The proof follows closely along the lines of [20, Lem. 10.9]. First, note that if $|\mathcal{S}| < \infty$, the assertion follows from the first part of the proof of Lemma 4.6. So henceforth we assume that $\mathcal{S}$ is a countably infinite set. For every $k \in \mathcal{S}$ we have

$$f(x_k) - f(x_{k+1}) \geq \eta_0 \left(m_k^f(x_k) - m_k^f(x_{k+1}) \right) \geq \eta_0 \mu_1 \alpha_k(\rho_k)\rho_k,$$

where we used Assumption 3.2(a) in the second inequality. Due to STEP 1 in Algorithm 1 we have $\alpha_k(\rho_k) \geq \min\{\varepsilon_c, \mu^{-q}\rho_k^q\}$, yielding

$$f(x_k) - f(x_{k+1}) \geq \eta_0 \mu_1 \min\{\varepsilon_c, \mu^{-q}\rho_k^q\}\rho_k. \tag{4.6}$$

Since $\mathcal{S}$ is countably infinite and the objective function f is bounded from below within the feasible set X, the right-hand side of (4.6), i.e., the trust region radius ρ_k, has to converge to zero. $\square$

Lemma 4.7 shows that using the stopping criterion $\rho_{k+1} < \rho_{min}$ is reasonable and results in termination of NOWPAC after a finite number of steps. Another direct consequence of Lemma 4.7 is that

$$\liminf_{k\to\infty} \alpha_k(\rho_k) = 0, \tag{4.7}$$

since $\alpha_k(\rho_k) \geq \kappa_1$ for some $\kappa_1 > 0$ for all k implies $\rho_k \geq \kappa_2$ for all k. The following theorem shows that the convergence of a subsequence of the approximated criticality measures $\{\alpha_{k_i}\}_{k_i}$ is carried over to the overall convergence of the exact criticality measures to zero.

THEOREM 4.8. *It holds that*

$$\lim_{k\to\infty} \mathfrak{A}[x_k] = 0.$$

Proof. Since the theorem holds for $|\mathcal{S}| < \infty$ by Lemma 4.6, we assume that $\mathcal{S}$ is a countably infinite set. Following the ideas of the proof of [20, Thm. 10.13] we prove the assertion of the theorem by contradiction. Assume that there exists a subsequence $\{\hat{k}_i\}_i \subseteq \mathcal{S}$ such that

$$\mathfrak{A}[x_{\hat{k}_i}] \geq \varepsilon_0 \tag{4.8}$$

for some $\varepsilon_0 > 0$ for all i. It immediately follows from Lemma 4.2 that

$$\alpha_{\hat{k}_i}(\rho_{\hat{k}_i}) \geq \varepsilon$$

for some $\varepsilon > 0$ for all i sufficiently large; in particular this holds true for

$$\varepsilon < \frac{\varepsilon_0}{4}. \tag{4.9}$$

13

Based on the subsequence $\{\hat{k}_i\}_i$ we define two subsequences $\{k_i\}_i$ and $\{l_i\}_i$ of all steps as follows: starting from $k_1 = \hat{k}_1$ we choose the first index $l_1 > k_1$ for which $\alpha_{l_1}(\rho_{l_1}) < \varepsilon$ and define the remaining members of the two subsequences inductively. Determine $j_i := \min\{j \in \mathbb{N} : \hat{k}_j > l_i\}$, set $k_{i+1} = \hat{k}_{j_i}$, and choose $l_{i+1} > k_{i+1}$ as being the first index for which $\alpha_{l_{i+1}}(\rho_{l_{i+1}}) < \varepsilon$. Note that the existence of $\{l_i\}_i$ is guaranteed by (4.7). We thus arrive at subsequences of indices satisfying

$$\alpha_k(\rho_k) \geq \varepsilon \quad \text{for} \quad k_i \leq k < l_i \quad \text{and} \quad \alpha_{l_i}(\rho_{l_i}) < \varepsilon. \tag{4.10}$$

Before we conclude the proof of convergence of $\{x_k\}_k$ to a first-order stationary point, we first have to show that $\lim_{i \to \infty} \|x_{k_i} - x_{l_i}\| = 0$; cf. the proof of [19, Thm. 10.13]. For this we define the set of indices

$$\mathcal{K} := \bigcup_{i \in \mathbb{N}_0} \{k \in \mathbb{N}_0 : k_i \leq k < l_i\},$$

with the sequences $\{k_i\}_i$ and $\{l_i\}_i$ as defined above. We know that $\alpha_k(\rho_k) \geq \varepsilon$ for $k \in \mathcal{K}$. Thus, since $\rho_k \to 0$ (see Lemma 4.7) it follows from Lemma 4.4 that the iteration k is acceptable or successful for all $k \in \mathcal{K}$ large enough. For every $k \in \mathcal{K} \cap \mathcal{S}$ we have

$$f(x_k) - f(x_{k+1}) \geq \eta_0 \left(m_k^f(x_k) - m_k^f(x_{k+1}) \right) \geq \eta_0 \mu_1 \alpha_k(\rho_k) \rho_k \geq \eta_0 \mu_1 \varepsilon \rho_k.$$

Thus we obtain

$$\|x_{k_i} - x_{l_i}\| \leq \sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{l_i - 1} \|x_j - x_{j+1}\| \leq \sum_{\substack{j=k_i \\ j \in \mathcal{K} \cap \mathcal{S}}}^{l_i - 1} \rho_j \leq \frac{1}{\eta_1 \mu_1 \varepsilon} \left(f(x_{k_i}) - f(x_{l_i}) \right)$$

for k_i sufficiently large. Noting that the sequence $\{f(x_k)\}_k$ is bounded from below (see Assumption 2.2) and monotonically decreasing, it follows that the left-hand side of the inequality above must converge to zero for $i \to \infty$.

Now, due to the continuity of the exact criticality measure (see Lemma A.3) and $\|x_{k_i} - x_{l_i}\| \to 0$, it holds that $|\mathfrak{A}[x_{l_i}] - \mathfrak{A}[x_{k_i}]| < \varepsilon$ for i sufficiently large. Moreover, using the fact that $\rho_{l_i} \to 0$, Corollary A.5 along with Lemma A.4 yields

$$\mathfrak{A}[x_{k_i}] \leq |\mathfrak{A}[x_{k_i}] - \mathfrak{A}[x_{l_i}]| + |\mathfrak{A}[x_{l_i}] - \mathfrak{A}_1[x_{l_i}]| + |\mathfrak{A}_1[x_{l_i}] - \mathfrak{A}_2[x_{l_i}]| + \mathfrak{A}_2[x_{l_i}]$$
$$< \varepsilon + \varepsilon + \varepsilon + \alpha_{l_i}(x_{l_i}) < 4\varepsilon < \varepsilon_0$$

for i sufficiently large, which contradicts (4.8). $\square$

5. Implementation and choice of parameters. Having discussed the theoretical properties of Algorithm 1 in Section 4, we now comment on the practical implementation of NOWPAC. In particular we address the practical choice of the order reduction parameter p as well as the existence of trial steps $\{s_k\}$ in STEP 2 satisfying Assumptions 3.2.

First we examine the existence of trial steps satisfying Assumptions 3.2; for this we consider the optimal solutions $\{\hat{d}_k\}$ of the criticality subproblem (3.6). We assume that the refinements in STEP 1 result in $\|g_k^f\| > 0$ eventually; otherwise, since STEP 1 ensures full linearity of the objective model, we have $\|\nabla f(x_k)\| = 0$, i.e., x_k is already a first-order critical point. It holds that

$$\min \left\{ \varepsilon_c, \, \mu^{-q} \rho_k^q \right\} \leq \alpha_k(\rho_k) = -\frac{1}{\rho_k} \left\langle g_k^f, \hat{d}_k \right\rangle = \frac{1}{\rho_k} \left\| g_k^f \right\| \left\| \hat{d}_k \right\| \cos(\phi_d),$$

where ϕ_d denotes the angle between $-g_k^f$ and $\hat{d}_k$. The first inequality is a direct consequence of STEP 1 in Algorithm 1 for ρ_k sufficiently small. Moreover, since $\hat{d}_k$ is a descent direction we know that

14

$\phi_d < \frac{\pi}{2}$ and thus $\cos(\phi_d) > 0$. Thus, for every x_k that is not an optimal solution of (1.1) we have

$$\left\|\hat{d}_k\right\| \geq \frac{\min\left\{\varepsilon_c,\, \mu^{-q}\rho_k^q\right\}\rho_k}{\left\|g_k^f\right\|\cos(\phi_d)},$$

which justifies Assumption 3.2(b). For Assumption 3.2(a) we note that $m_k^f(x_k) - m_k^f(x_k + \hat{d}_k) = -\langle g_k^f, \hat{d}_k\rangle - t(\hat{d}_k)$ with the remainder term of the Taylor approximation $t(\hat{d}_k) \in \mathcal{O}(\|\hat{d}_k\|^2)$. It holds that

$$m_k^f(x_k) - m_k^f(x_k + \hat{d}_k) = -\left\langle g_k^f, \hat{d}_k\right\rangle - t(\hat{d}_k) \geq \rho_k\alpha_k(\rho_k) - \left|t(\hat{d}_k)\right|$$

$$= \rho_k\alpha_k(\rho_k) - \|\hat{d}_k\|^2\left|t(\hat{d}_k)\|\hat{d}_k\|^{-2}\right|$$

$$\geq \rho_k\alpha_k(\rho_k) - \rho_k^2\left|t(\hat{d}_k)\|\hat{d}_k\|^{-2}\right| \geq \frac{1}{2}\rho_k\alpha_k(\rho_k),$$

for ρ_k sufficiently small. For the last inequality we used the fact that $\rho_k \in o(\alpha_k(\rho_k))$, which is ensured by STEP 1 in Algorithm 1. In our implementation of NOWPAC, however, we avoid repeating STEP 1 whenever the trial step s_k is infeasible and go directly to line 10 in Algorithm 1. We do this to reduce the computational costs of computing the criticality measure at every infeasible step.

NOWPAC is designed to work in settings with costly objective function evaluations that dominate the cost of computing a good trial step s_k. This suggests it may be beneficial (in terms of the overall computational costs) to invest effort in computing a good trial step rather than looking for a quick and crude approximation. In our implementation of NOWPAC we use the CCSA algorithm [63], as implemented in the NLopt library [32], to compute the trial steps in STEP 2 of Algorithm 1, i.e., to find $s_k = \arg\min_{x_k + s \in X_k,\, \|s\| \leq \rho_k} m_k^f(x_k + s)$.

Next we discuss the choice of the order reductions p and q. We briefly recall where we introduced p and q:

- The order reduction parameter p appears in the definition of the inner boundary path (3.1); it is required within Lemma 4.3 to prove that STEP 3 of Algorithm 1 eventually finds a feasible trial step, when the trust region radius is small enough.
- Within Lemma 4.3, we used Assumption 3.2(b) on the step size being a fraction of order ρ_k^{1+q}; the latter assumption is ensured by STEP 1 of Algorithm 1.

In practical applications, Algorithm 1 is always terminated when the trust region radius falls below the threshold ρ_{min}. We therefore discuss the choice of order reductions $p = 0$ and $q = 0$ in the *pre-asymptotic regime* of $\rho_k \geq \rho_{min}$.

We note that, as long as STEP 2 computes a descent direction s_k, we can always find (potentially small) parameters μ_1 and μ_2 such that Assumption 3.2 is satisfied with $q = 0$ for all $\rho_k \geq \rho_{min}$. Thus, Assumption 3.2 can be satisfied for $\rho_k \geq \rho_{min}$, regardless of the choice of q in STEP 1 of Algorithm 1, allowing us simply to check for $\rho_k > \mu\alpha_k(\rho_k)$ in STEP 1 of Algorithm 1. Revisiting the proof of Lemma 4.3 with $p = 0$ and using $\|s_k\| \geq \mu_2\rho_k^1$, we see that STEP 3 computes a feasible trial step s_k if $\varepsilon_b \geq \kappa_c\mu_2^{-2}$. Thus we are guaranteed to find feasible trial steps by choosing the inner boundary path constant large enough, even for the choice of $p = 0$ in the definition of the inner boundary path. Finally, note that choosing the inner boundary path to be a quadratic offset ($p = 0$) is consistent with Assumption 3.1 and Lemma A.4 since its Hessian is positive definite. Finally, we propose a heuristic for an adaptive choice of the inner boundary path constant ε_b. From Lemma 4.3 we see that ε_b has to be chosen sufficiently large in order to convexify the constraints and to guarantee that STEP 3 in Algorithm 1 will be passed with a feasible trial step. In practical applications, however, we have found that NOWPAC works very well for an *a priori* fixed value of $\varepsilon_b > 0$ along with the adaptive scaling

$$\varepsilon_{b,k} := \varepsilon_b\left(\frac{\|s_{k-1}\|}{\rho_{k-1}}\right)^2, \tag{5.1}$$

to increase efficiency by not overly constraining the trial steps due to the inner boundary path. The situation of a too-large inner boundary path constant and its relaxation (5.1) is depicted in the left plot of Figure 3.1, where we see an unnecessary restriction of the possible step size.

6. Inexact evaluations of the objective function and constraints. In the preceding sections we assumed that we are able to evaluate the objective function and the constraints up to a prescribed tolerance, so that the models m_k^f and m_k^c are fully linear (2.1). This assumption requires the function evaluations to become more and more accurate when approaching a critical point. As we noted in Section 1, there exist theoretical results in the context of derivative-based trust region methods (see [14, 29] and the references therein) showing convergence in case of increasing accuracy of the evaluations while approaching the optimal design. For corresponding results for derivative-free methods, see for example [12, 35]. In practical applications, we are often faced with situations where we cannot avoid inexact evaluations of the objective function or the constraints. Inexactness may stem from numerical errors, limitations on the number of cycles in a recursive procedure, inaccurate measurements, and other factors. Particularly in cases where the objective function and constraints are given only as black-box evaluations of a simulation code, we are not likely to be able to tune the model tolerances. Figure 7.3 provides an example of the inexact function evaluations that we would like our method to address; shown are evaluations of the objective function and a constraint function in the tar removal process model of Section 7.4. The small-scale roughness is the result of numerical errors.

To avoid any ambiguity, we contrast our focus on numerical errors with the case of an objective or constraint function that depends on uncertain parameters, where the parameters may be constrained to some interval or endowed with a probability distribution. In the latter case, one might account for uncertainty by replacing the objective function or constraint with its "robust counterpart," yielding a task in stochastic programming. We refer the interested reader to [10, 11, 36, 55, 60] and references therein. Methods for stochastic programming require the exploration of the uncertain parameter space in some fashion, and are not our focus here. Of course, there is a link between the introduction of robust objectives and the issue of numerical error; for instance, numerical evaluation of an expectation with respect to the uncertain parameters is subject to error due to a finite number of Monte Carlo samples or finite quadrature resolution. But our focus here is on the presence and magnitude of numerical errors only, regardless of how they originated. In other words, we do not distinguish among different sources of inexactness in evaluations of f and c.

This section first addresses the situation where increasing the accuracy of the evaluations of the objective function and the constraints is possible. In this case we quantify the rate of noise reduction needed to guarantee convergence. Thereafter, we discuss regimes where the accuracy level cannot be adjusted and propose an indicator to detect when inexact evaluations of f or c prevent NOWPAC from making progress. In this case, we propose early termination of the algorithm to save computational effort and to prevent corruption of the results.

For the error analysis in this section we assume that the objective function and the constraints can each be split into a sum of two terms. The first terms are the functions themselves, satisfying Assumptions 2.2 and 2.3. The second terms are the errors. These error terms are only observed (via summation with the exact function value) at the finite number of points where the objective function and the constraints are evaluated. We fill in the gaps between these points using quadratic extensions of the error, via minimum-Frobenius-norm models δ_k^f and δ_k^c. We point out that δ_k^f and δ_k^c are simply extensions of the observed errors, rather than approximations of the actual error. We assume that the magnitudes of the errors are bounded by $\delta_{k,max}^f$ and $\delta_{k,max}^c$. Beyond this, we do not make any additional assumptions, e.g., on the distribution of the error or even whether it is stochastic or deterministic. In order to be detectable, however, the errors at different points in the design space must be sufficiently uncorrelated. For example, if the error term degenerates to a constant offset, it is impossible to separate it from the underlying objective function or constraint by simply observing its sum with one of the latter. The same holds true for errors satisfying equivalent

smoothness properties as the objective function and the constraints. Summarizing, the perturbed observed functions are given by

$$f_{n,k}(x) := f(x) + \delta_k^f(x),$$
$$c_{n,k}(x) := c(x) + \delta_k^c(x).$$

In the following theorem we prove the rate of decay—with respect to the trust region radii—that the errors δ_k^f and δ_k^c must obey in order to guarantee convergence of NOWPAC; cf. also [34]. To simplify notation in the following theorem, without loss of generality we assume the design variables to be properly scaled and set $\rho_{max} = 1$.

THEOREM 6.1. *Consider the fully linear minimum-Frobenius-norm surrogates $m_k^{f_{n,k}}$ and $m_k^{c_{n,k}}$ of the observed noisy objective function $f_{n,k}$ and constraints $c_{n,k}$. The intermediate points $\{x_k\}_k$ computed by Algorithm 1 converge to a first-order critical point if $\delta_{k,max}^f, \delta_{k,max}^c \in o\left(\rho_k^2\right)$ and the inner boundary path constant ε_b is greater than $\mu_2^{-\frac{2}{1+p}}(\kappa_c + \delta_{k,max}^c \rho_k^{-2\frac{1+q}{1+p}})$.*

Proof. First we show that if $\delta_{k,max}^f, \delta_{k,max}^c \in o\left(\rho_k^2\right)$, the full linearity of the noisy models is maintained. In [20, Thm. 5.4] it is shown that for minimum-Frobenius-norm models we have

$$\left| f_{n,k}(x_k + s) - m_k^{f_{n,k}}(x_k + s) \right| \leq \kappa_{f_n} \rho_k^2$$
$$\left| c_{n,k}(x_k + s) - m_k^{c_{n,k}}(x_k + s) \right| \leq \kappa_{c_n} \rho_k^2$$
$$\left\| \nabla f_n(x_k + s) - \nabla m_k^{f_n}(x_k + s) \right\| \leq \kappa_{df_n} \rho_k$$
$$\left\| \nabla c_n(x_k + s) - \nabla m_k^{c_n}(x_k + s) \right\| \leq \kappa_{dc_n} \rho_k,$$

with

$$
\begin{aligned}
\kappa_{f_n} &= \left(\kappa_n + \frac{1}{2} \right) \left(\nu_{n,k}^f + \left\| H_k^{f_{n,k}} \right\| \right) \\
\kappa_{c_n} &= \left(\kappa_n + \frac{1}{2} \right) \left(\nu_{n,k}^c + \left\| H_k^{c_{n,k}} \right\| \right) \\
\kappa_{df_n} &= \kappa_n \left(\nu_{n,k}^f + \left\| H_k^{f_{n,k}} \right\| \right) \\
\kappa_{dc_n} &= \kappa_n \left(\nu_{n,k}^c + \left\| H_k^{c_{n,k}} \right\| \right).
\end{aligned}
\tag{6.1}
$$

Here the constant κ_n depends on the geometry of the interpolation points, but it does not depend on the trust region radius ρ_k. Also, $\nu_{n,k}^f$ and $\nu_{n,k}^c$ denote the Lipschitz constants of the gradients of $f_{n,k}$ and $c_{n,k}$, while $H_k^{f_{n,k}}$ and $H_k^{c_{n,k}}$ denote the Hessians of $m_k^{f_{n,k}}$ and $m_k^{c_{n,k}}$. Now we examine the Lipschitz constants of the gradients of $f_{n,k}$ and $c_{n,k}$. Using the triangle inequality we get

$$\|\nabla f_{n,k}(x_1) - \nabla f_{n,k}(x_2)\| \leq \|\nabla f(x_1) - \nabla f(x_2)\| + \left\| \nabla \delta_k^f(x_1) + \nabla \delta_k^f(x_2) \right\|$$
$$\leq \left(\nu^f + \left\| H_k^{\delta_k^f} \right\| \right) \|x_1 - x_2\|,$$
$$\|\nabla c_{n,k}(x_1) - \nabla c_{n,k}(x_2)\| \leq \|\nabla c(x_1) - \nabla c(x_2)\| + \left\| \nabla \delta_k^c(x_1) + \nabla \delta_k^c(x_2) \right\|$$
$$\leq \left(\nu^c + \left\| H_k^{\delta_k^c} \right\| \right) \|x_1 - x_2\|,$$

where ν^f and ν^c are the Lipschitz constants of ∇f and ∇c, and $H_k^{\delta_k^f}$ and $H_k^{\delta_k^c}$ denote the Hessians of the error functions $\delta_k^f(x)$ and $\delta_k^c(x)$. From the above inequalities we obtain upper bounds on the

Lipschitz constants:

$$\nu_{n,k}^f \leq \nu^f + \left\| H_k^{\delta_k^f} \right\| \quad \text{and} \quad \nu_{n,k}^c \leq \nu^c + \left\| H_k^{\delta_k^c} \right\|. \tag{6.2}$$

Furthermore it holds that

$$\left\| H_k^{f_{n,k}} \right\| = \left\| H_k^f + H_k^{\delta_k^f} \right\| \leq \left\| H_k^f \right\| + \left\| H_k^{\delta_k^f} \right\| \quad \text{and}$$

$$\left\| H_k^{c_{n,k}} \right\| = \left\| H_k^c + H_k^{\delta_k^c} \right\| \leq \| H_k^c \| + \left\| H_k^{\delta_k^c} \right\|,$$

which, together with (6.2) and (6.1), yields

$$
\begin{aligned}
\kappa_{f_n} &\leq \left(\kappa_n + \frac{1}{2} \right) \left(\nu^f + \left\| H_k^f \right\| + 2 \left\| H_k^{\delta_k^f} \right\| \right) \leq \kappa_1^f + 2\kappa_2^f \delta_{k,max}^f \rho_k^{-2} \\
\kappa_{c_n} &\leq \left(\kappa_n + \frac{1}{2} \right) \left(\nu^c + \| H_k^c \| + 2 \left\| H_k^{\delta_k^c} \right\| \right) \leq \kappa_1^c + 2\kappa_2^c \delta_{k,max}^c \rho_k^{-2} \\
\kappa_{df_n} &\leq \kappa_n \left(\nu^f + \left\| H_k^f \right\| + 2 \left\| H_k^{\delta_k^f} \right\| \right) \leq \kappa_3^f + \kappa_2^f \delta_{k,max}^f \rho_k^{-2} \\
\kappa_{dc_n} &\leq \kappa_n \left(\nu^c + \| H_k^c \| + 2 \left\| H_k^{\delta_k^c} \right\| \right) \leq \kappa_3^c + \kappa_2^c \delta_{k,max}^c \rho_k^{-2}.
\end{aligned}
\tag{6.3}
$$

for constants $\kappa_1^f, \kappa_1^c, \kappa_2^f, \kappa_2^c, \kappa_3^f, \kappa_3^c > 0$. In the second inequalities we used

$$\left\| H_k^{\delta_k^f} \right\| \leq \bar{\kappa}_2^f \frac{\delta_{k,max}^f}{\rho_k^2} \quad \text{and} \quad \left\| H_k^{\delta_k^c} \right\| \leq \bar{\kappa}_2^c \frac{\delta_{k,max}^c}{\rho_k^2},$$

cf. [20, Thm. 5.7] where we replace the upper bound mentioned in the proof of [20, Thm. 5.7] by $\max_{x \in B(x_k,\rho_k)} |\delta_k^f(x)| \leq \delta_{k,max}^f$ and $\max_{x \in B(x_k,\rho_k)} |\delta_k^c(x)| \leq \delta_{k,max}^c$ respectively. Thus the full linearity properties (2.1) and Assumption 2.3 hold, if we ensure that the right-hand sides of (6.3), i.e., the values of κ_{f_n}, κ_{c_n}, κ_{df_n} and κ_{bhf_n}, do not grow unboundedly; in particular this is the case if

$$\delta_{k,max}^f \in o\left(\rho_k^2 \right) \quad \text{and} \quad \delta_{k,max}^c \in \left(\rho_k^2 \right).$$

To conclude the proof of convergence, we relate the noisy function evaluations $f_{n,k}$ and $c_{n,k}$ to the exact objective function f and constraints c, i.e.,

$$
\begin{aligned}
|f(x_k + s) - f_{n,k}(x_k + s)| &\leq \delta_{k,max}^f \in o(\rho_k^2) \\
|c(x_k + s) - c_{n,k}(x_k + s)| &\leq \delta_{k,max}^c \in o(\rho_k^2) \\
\|\nabla f(x_k + s) - \nabla f_{n,k}(x_k + s)\| &= \left\| \nabla \delta_k^f(x_k + s) \right\| \in o(\rho_k) \\
\|\nabla c(x_k + s) - \nabla c_{n,k}(x_k + s)\| &= \| \nabla \delta_k^c(x_k + s) \| \in o(\rho_k).
\end{aligned}
$$

The latter inclusion follows from [19, Thm. 5.4] by interpreting δ_k^f, or δ_k^c respectively, as an $o(\rho_k^2)$ approximation of the constant zero function and replacing the assumption on exact function evaluations in the proof of [19, Thm. 5.4] with the point-wise error $o(\rho_k^2)$ (see Lemma A.6 for more details). In summary, $m_k^{f_{n,k}}$ and $m_k^{c_{n,k}}$ are fully linear models for the exact objective function f and the constraints c, respectively.

Finally we address the issue that Algorithm 1 may pass STEP 3 with a trial step that is incorrectly designated as feasible. We have to ensure that no infeasible step is accepted because it appears to

be feasible due to the noise. For this we revisit the proof of Lemma 4.3 and now require the safety margin $c(x_k + s_k) \leq -\delta^c_{k,max}$. It follows that

$$\rho_k \leq \left(\frac{1}{\kappa_c} \left(\varepsilon_b \mu_2^{\frac{2}{1+p}} - \delta^c_{k,max} \rho_k^{-2\frac{1+q}{1+p}} \right) \right)^{\frac{1+p}{2(p-q)}}.$$

Thus, in the pre-asymptotic phase we have to choose ε_b large enough so that the right-hand side in the inequality above is always greater than $\rho_{max} = 1$, yielding

$$\varepsilon_b \geq \mu_2^{-\frac{2}{1+p}} \left(\kappa_c + \delta^c_{k,max} \rho_k^{-2\frac{1+q}{1+p}} \right),$$

where this additional restriction vanishes for decreasing trust region radius ρ_k. $\Box$

Note that even though we only have access to inexact evaluations of the constraints, we still have to be able to check feasibility in STEP 3 in Algorithm 1. However, we point out that in the asymptotic regime $k \to \infty$, $\delta^c_{k,max} = o(\rho_k^2)$ will be dominated by the requirement from Assumption 3.1. Thus, in practical applications, the inner boundary path constant ε_b has to be adapted to the magnitude of the errors (or simply chosen sufficiently large) in order to ensure convergence of Algorithm 1 also in the pre-asymptotic regime where the trust region radii are not close to zero.

We now use Theorem 6.1 to define an indicator that estimates the minimum trust region radius at which further progress of Algorithm 1 is expected not to improve the optimization result. In other words, the indicator detects regimes in which $\delta^f_{k,max}, \delta^c_{k,max} = o\left(\rho_k^2\right)$ is violated. The indicator is based on observing the increase of the norms of $H_k^{f_n}$ or $H_k^{c_n}$ as functions of the trust region radius. Note that these Hessians are computed in every iteration of Algorithm 1 and are therefore readily available without additional computational cost. As described in the proof of Theorem 6.1, the norms of $H_k^{f_n}$ and $H_k^{c_n}$ show the same asymptotic behavior as $\delta^f_{k,max}\rho_k^{-2}$ and $\delta^c_{k,max}\rho_k^{-2}$ for decreasing trust region radii. Thus, on a logarithmic scale, the slope τ of the growth of $\|H_k^{f_n}\|$ and $\|H_k^{c_n}\|$ with respect to the trust region radii ρ_k should not grow for vanishing ρ_k. We estimate this slope using linear regression of the corresponding norms of the Hessians at rejected steps, i.e., at steps where the intermediate point does not change. In these steps, the Hessian is supposed to be of the same order of magnitude, provided the geometry constant κ_n does not grow unboundedly. The latter property is ensured by Algorithm 1 within the model improvement steps. The result is the approximated slope $\tau\left(\|H_k\|\right)$, where H_k denotes either $H_k^{f_n}$ or $H_k^{c_n}$. The slope can be categorized as indicating a convergent or possibly non-convergent regime, according to the corresponding convergence properties of Algorithm 1; in our numerical examples we set the noise threshold to 1, i.e., we used the classification $\tau\left(\|H_k\|\right) < 1$ for the convergent regime and $\tau\left(\|H_k\|\right) \geq 1$ for the possibly non-convergent regime. If possibly non-convergent iterations are detected, we suggest early termination of Algorithm 1 in order to prevent deterioration of the results.

7. Numerical results. In this section we apply NOWPAC to several optimization problems. In Sections 7.1 and 7.2, we discuss two model problems: the Rosenbrock function (7.1) and a nonlinear constrained anisotropic exponential example (7.2). We use these examples for validation of the algorithm, knowing the exact optimal points and the minimum objective values. In both examples, we also demonstrate the effectiveness of the error indicator proposed in Section 6. In Section 7.3 we demonstrate NOWPAC's efficiency on test problems from the Schittkowski benchmark set [59]. Then in Section 7.4, we apply NOWPAC to a large-scale black box model of tar removal in a biomass-to-liquid plant. In this example, the model enters evaluations of both the objective and the constraints. In all examples, we set the parameters in NOWPAC to the values shown in Table 7.1. We use the initial trust region radius $\rho_0 = 0.1$ throughout the examples.

For comparison we also compute the optimal points using the linear surrogate-based solver COBYLA and the direct search algorithms NOMAD, SDPEN, and GSS-NLC. We use the implementation of COBYLA [46, 47] from the NLopt optimization library for nonlinear optimization [32]

TABLE 7.1
Default parameters of NOWPAC used in the test examples of Sections 7.1–7.4.

parameter	ε_b	γ_{inc}	γ	ω	η_0	η_1
value	10.0	2.0	0.8	0.6	0.1	0.7

and the implementation of NOMAD as in [37]. The asynchronous parallel pattern search (APPS) method GSS-NLC [2, 28] is a pattern search method based on sampling of the objective function augmented by a penalty approach for the constraints. In all examples we picked an initial penalty parameter of 10^3 and an initial step size of 0.1, where we found GSS-NLC to perform best. We used the GSS-NLC code as provided by the hybrid optimization parallel search package (HOPSPACK) [44]. All computations for the benchmark problems were performed on a 2.6 GHz Intel Core i7 processor using the GNU compiler version 4.8.

7.1. Rosenbrock function. The first example is the unconstrained optimization of the Rosenbrock function,

$$\min_{(x_1,x_2)\in\mathbb{R}^2} (x_2 - x_1^2)^2 + (x_1 - 1)^2, \tag{7.1}$$

where the optimal point $x^* = (1,1)^T$ and the minimal objective value $f^* = 0$ are known analytically. We start the optimization at $x_0 = (1.5, 1.5)^T$. The Rosenbrock function exhibits small gradients in the neighborhood of the optimal point x^*, and thus constitutes a setting in which derivative-free trust region methods like NOWPAC are forced to take small steps; we recall that the size of the trust region, and therefore the step size, is tightly connected to the size of the gradients. For this reason, we find it worthwhile to include this unconstrained test case to discuss the performance of NOWPAC.

The performances of the solvers NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC are summarized in Table 7.2. We see that all methods result in reasonable approximations of the exact optimum. Looking at the number of function evaluations, we see that NOWPAC requires fewer function evaluations than the other solvers.

Next, we introduce artificial errors into the objective function. In particular, we add independent errors to every evaluation of the objective function, randomly drawn from a uniform distribution on the interval $[-\delta_{max}^f, \delta_{max}^f]$ with magnitude $\delta_{max}^f \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. We show the norms of the Hessians H_k^f for one realization of an optimization run of NOWPAC in Figure 7.1. To create
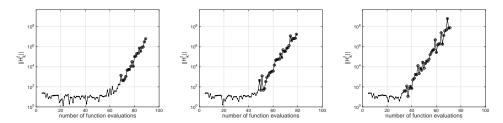

FIG. 7.1. *Norms of the Hessians of the models of the objective function in (7.1) for $\delta_{max}^f = 10^{-4}$ (left), $\delta_{max}^f = 10^{-3}$ (middle) and $\delta_{max}^f = 10^{-2}$ (right). Circles indicate the non-convergent regime.*

these plots we switch off the early termination due to the detection of errors. (Note that NOWPAC would ordinarily stop after detecting a non-convergent iteration, as described in Section 6.) In Table 7.3 we report the average distance from the approximated optimal point to the exact optimal design at early termination, as well as the corresponding average distances for the objective values. To compute these averages we ran NOWPAC 1000 times with random samples for the errors in

TABLE 7.2
Summarized performance statistics of NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC applied to the Rosenbrock minimization problem 7.1. SC indicates the stopping criteria, which is the ρ_{min} threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA, NOMAD, SDPEN, and GSS-NLC. d_x and d_f denote the Euclidean distances between the approximated and the analytical solution.

	SC	#eval	d_x	d_f
NOWPAC	10^{-3}	64	$2.27 \cdot 10^{-4}$	$1.05 \cdot 10^{-8}$
COBYLA	10^{-3}	81	$1.81 \cdot 10^{-2}$	$6.29 \cdot 10^{-5}$
NOMAD	10^{-3}	70	0	0
SDPEN	10^{-3}	69	0	0
GSS-NLC	10^{-3}	129	$6.12 \cdot 10^{-3}$	$7.24 \cdot 10^{-6}$
NOWPAC	10^{-4}	65	$2.27 \cdot 10^{-4}$	$1.05 \cdot 10^{-8}$
COBYLA	10^{-4}	150	$8.47 \cdot 10^{-4}$	$2.81 \cdot 10^{-7}$
NOMAD	10^{-4}	81	0	0
SDPEN	10^{-4}	85	0	0
GSS-NLC	10^{-4}	184	$4.07 \cdot 10^{-4}$	$3.47 \cdot 10^{-8}$
NOWPAC	10^{-5}	76	$1.09 \cdot 10^{-4}$	$2.07 \cdot 10^{-9}$
COBYLA	10^{-5}	199	$1.05 \cdot 10^{-4}$	$2.37 \cdot 10^{-9}$
SDPEN	10^{-5}	97	0	0
NOMAD	10^{-5}	97	0	0
GSS-NLC	10^{-5}	228	$6.35 \cdot 10^{-5}$	$7.28 \cdot 10^{-10}$

the objective function. All distances are computed at the iteration where NOWPAC detects the first non-convergent iteration. In the same table, we report the average number of saved function evaluations, i.e., the number of additional evaluations performed by a run with identical parameters but *without* early termination due to inexact function evaluations. The numbers are rounded to the nearest integer. As expected, increasing the magnitude of the errors corrupts the optimal

TABLE 7.3
Summarized performance of the application of NOWPAC to the noisy Rosenbrock minimization problem. The stopping criteria is set to $\rho_{min} = 10^{-5}$. d_x and d_f denote the average Euclidean distances of the approximations to the analytical solutions at early termination. d_x^{end} and d_f^{end} denote the average Euclidean distances of the approximations to the analytical solutions without early termination. The solution is computed 1000 times and the average value of all outcomes is shown. #eval refers to the average number of function evaluations at early termination, whereas #saved is the average number of additional evaluations performed by the same run without early termination.

δ_{max}^f	#eval	#saved	d_x	d_f	d_x^{end}	d_f^{end}
10^{-2}	33	33	$2.66 \cdot 10^{-1}$	$1.01 \cdot 10^{-2}$	$2.64 \cdot 10^{-1}$	$9.48 \cdot 10^{-3}$
10^{-3}	56	30	$5.38 \cdot 10^{-2}$	$7.75 \cdot 10^{-4}$	$5.33 \cdot 10^{-2}$	$7.79 \cdot 10^{-4}$
10^{-4}	71	22	$1.63 \cdot 10^{-2}$	$8.75 \cdot 10^{-5}$	$1.60 \cdot 10^{-2}$	$8.60 \cdot 10^{-5}$
10^{-5}	80	16	$5.18 \cdot 10^{-3}$	$9.01 \cdot 10^{-6}$	$4.98 \cdot 10^{-3}$	$8.59 \cdot 10^{-6}$
0	76	–	–	–	$1.09 \cdot 10^{-4}$	$2.07 \cdot 10^{-9}$

point. Moreover, the number of objective function evaluations declines significantly, even if early termination is switched off. To explain this trend, we point out that d_f is roughly of the same order as the maximal magnitude of the errors, δ_{max}^f. In this situation, the inexact function evaluations corrupt the acceptance ratio r_k in STEP 4 of Algorithm 1, misleading NOWPAC to reject steps. Using the noise indicator we are able to detect this regime and automatically terminate NOWPAC without evaluating many steps that will eventually be rejected. We should also point out that choosing a stopping criterion ρ_{min} that is *a priori* adjusted to the magnitude of the noise would

also avoid many rejected steps, due to the noise overwhelming the shape of the objective. But in the examples above we did not presume to know anything about the magnitude of the noise, i.e., we did not change parameters of the algorithm according to δ_{max}^f. This setup is intended to mimic practical situations in which the noise is not well characterized, and hence manually tailoring the stopping criterion to the noise magnitude is not possible.

7.2. Constrained anisotropic exponential. Our second example is the constrained minimization problem

$$\min_{x \in X} -\exp\left(x^T D x\right) \tag{7.2}$$

with the diagonal scaling matrix $D = \mathrm{diag}(1, 2, 3, 4, 5)$ and the feasible domain

$$X = \left\{ x \in \mathbb{R}^5 \ : \ \sin\left(\|x\|^2\right) \leq \frac{1}{2}, \ \left\|x - \frac{3}{8}e_5\right\| \leq \frac{3}{8} \right\},$$

where $e_5 = (0, 0, 0, 0, 1)^T$ denotes the fifth canonical unit vector. Here we have 5 design parameters and 2 constraint functions. The optimal point is known to be $x^* = (0, 0, 0, 0, \sqrt{\arcsin(0.5)})^T \approx (0, 0, 0, 0, 0.724)^T$ with optimal value $f^* = -1.37 \cdot 10^1$. At the optimal point, the first constraint $c_1(x) = \sin(\|x\|^2) - 0.5$ is active, so that the optimal point is critical, $\mathfrak{A}[x^*] = 0$, but not stationary, $\|\nabla f(x^*)\| = 3.96 \cdot 10^2$. Despite this steep gradient at x^*, the objective function exhibits a relatively flat region around the origin, where the greatest descent can be achieved by varying the last coordinate. Thus, when starting at the point $x_0 = (0.1, 0.1, 0.1, 0.1, 0.1)^T$, reducing the objective function drives the intermediate points towards the boundary of the feasible domain X. Once the boundary is reached, further progress towards the minimum can only be made by moving *along* the boundary of X. The shape of this objective and feasible domain therefore constitute a useful setting in which to discuss the effectiveness of constraint handling in NOWPAC.

TABLE 7.4
Summarized performance statistics of NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC applied to the constrained minimization problem (7.2). SC indicates the stopping criteria, which is the ρ_{min} threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA, NOMAD, SDPEN, and GSS-NLC. d_x and d_f denote the Euclidean distances between the approximated and the analytical solutions.

	SC	#eval	d_x	d_f
NOWPAC	10^{-3}	59	$2.23 \cdot 10^{-3}$	$2.87 \cdot 10^{-4}$
COBYLA	10^{-3}	81	$2.50 \cdot 10^{-3}$	$4.54 \cdot 10^{-4}$
NOMAD	10^{-3}	574	$6.37 \cdot 10^{-2}$	$6.57 \cdot 10^{-2}$
SDPEN	10^{-3}	153	$4.18. \cdot 10^{-1}$	9.49
GSS-NLC	10^{-3}	1610	$1.51 \cdot 10^{-2}$	$3.70 \cdot 10^{-3}$
NOWPAC	10^{-4}	96	$3.67 \cdot 10^{-4}$	$4.86 \cdot 10^{-6}$
COBYLA	10^{-4}	130	$8.74 \cdot 10^{-5}$	$1.86 \cdot 10^{-6}$
NOMAD	10^{-4}	980	$4.10 \cdot 10^{-2}$	$2.52 \cdot 10^{-2}$
SDPEN	10^{-4}	192	$4.18 \cdot 10^{-1}$	9.49
GSS-NLC	10^{-4}	4073	$9.60 \cdot 10^{-3}$	$1.28 \cdot 10^{-3}$
NOWPAC	10^{-5}	128	$3.15 \cdot 10^{-5}$	$1.63 \cdot 10^{-8}$
COBYLA	10^{-5}	160	$1.83 \cdot 10^{-5}$	$1.14 \cdot 10^{-8}$
NOMAD	10^{-5}	2842	$1.94 \cdot 10^{-2}$	$5.28 \cdot 10^{-3}$
SDPEN	10^{-5}	221	$4.18. \cdot 10^{-1}$	9.49
GSS-NLC	10^{-5}	9321	$2.23 \cdot 10^{-3}$	$4.49 \cdot 10^{-4}$

The performances of NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC are summarized in Table 7.4. Almost all methods, except SDPEN which did not converge to a critical solution,

produce reasonable approximations of the optimal point and the corresponding minimal value of the objective function. However, for all stopping thresholds, NOWPAC requires the smallest number of evaluations of the objective function. In particular, GSS-NLC requires many function evaluations—first, because it must explore the design space in all five coordinate directions, and second, because it repeatedly explores the space while adaptively choosing a suitably high penalty parameter to meet the prescribed tolerances. Also NOMAD requires many function evaluations to find an approximate solution; the achieved accuracy is far less than the solution computed by NOWPAC and COBYLA. On this test example, SDPEN converged prematurely to a non-critical solution.

Next, we again introduce artificial errors of different magnitudes into the objective function and constraint function evaluations. We solve (7.2) 1000 times and report, in Table 7.5, the average number of function evaluations (for termination after the first noisy iteration), the average number of saved function evaluations, as well as the average absolute error in the computed design and the corresponding objective value. We see that for noise in the objective function in particular, the number of saved function evaluations is significant.

TABLE 7.5

Summarized performance of the application of NOWPAC to the noisy constrained minimization problem (7.2). The stopping criteria is set to $\rho_{min} = 10^{-5}$. d_x and d_f denote the average Euclidean distances of the approximations to the analytical solutions at early termination. d_x^{end} and d_f^{end} denote the average Euclidean distances of the approximations to the analytical solutions without early termination. The solution is computed 1000 times and the average value of all outcomes is shown. #eval refers to the average number of function evaluations at early termination, whereas #saved is the average number of additional evaluations (saved evaluations) performed by the same run without early termination.

δ_{max}^f	δ_{max}^c	#eval	#saved	d_x	d_f	d_x^{end}	d_f^{end}
10^{-2}	10^{-5}	62	21	$6.43 \cdot 10^{-2}$	$1.31 \cdot 10^{-1}$	$6.22 \cdot 10^{-2}$	$8.04 \cdot 10^{-2}$
10^{-3}	10^{-5}	75	5	$2.81 \cdot 10^{-2}$	$1.63 \cdot 10^{-2}$	$2.81 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$
10^{-2}	10^{-4}	60	11	$7.51 \cdot 10^{-2}$	$1.39 \cdot 10^{-1}$	$7.45 \cdot 10^{-2}$	$1.28 \cdot 10^{-1}$
10^{-3}	10^{-4}	62	4	$6.31 \cdot 10^{-2}$	$9.28 \cdot 10^{-2}$	$6.28 \cdot 10^{-2}$	$9.00 \cdot 10^{-2}$
10^{-3}	0	75	22	$2.09 \cdot 10^{-2}$	$9.96 \cdot 10^{-3}$	$2.02 \cdot 10^{-2}$	$8.32 \cdot 10^{-3}$
0	10^{-4}	62	4	$5.90 \cdot 10^{-2}$	$8.30 \cdot 10^{-2}$	$5.90 \cdot 10^{-2}$	$8.24 \cdot 10^{-2}$
0	0	128	—	—	—	$3.15 \cdot 10^{-5}$	$1.63 \cdot 10^{-8}$

7.3. Schittkowski benchmark problems. In this section we compare the efficiency of NOW-PAC with the derivative-free optimization codes COBYLA, NOMAD, SDPEN, and GSS-NLC on a broader set of benchmark problems. For this comparison we chose nonlinear constrained optimization problems from the Schittkowski benchmark problem collection for nonlinear programming [59]; see also [30, 58]. The number of design variables in these problems varies between 2 and 15, and the number of nonlinear constraints varies between 1 and 10. Results from all the optimization codes run with stopping criteria $SC = 10^{-3}$ and $SC = 10^{-5}$ are summarized in Tables 7.6 and 7.7. In all the test problems, the exact optimal design and the optimal value are known; thus we can report the absolute error d_x in the optimal designs as well as the absolute and relative errors, d_f^{abs} and d_f^{rel}, in the objective functions. We see that in many test examples, NOWPAC performs—in terms of required function evaluations—significantly better than the other optimization codes. Moreover, in all cases, the accuracy of the computed optimal solution is better than or comparable to that of the other codes.

Note that in some of the test problems, the direct search methods (i.e., NOMAD, SDPEN, and GSS-NLC) find the exact solution. In our benchmark set this is the case if the initial and optimal designs have simple integer values. Nevertheless, despite the fact that these codes find the optimal solutions relatively early, they need a significant number of function evaluations to certify the optimality of the solution. These codes' fast descent of the objective function unfortunately gets lost in many other cases, particularly the moderate-dimensional examples (test problems 100,

113, and 285). In contrast, NOWPAC consistently yields a good reduction of the objective even for increasing dimensionality of the optimization problem.

7.4. Tar removal process model. We now discuss optimization of a tar removal process, shown schematically in Figure 7.2, which is part of the production of synthesis gas (syngas) in a biomass to liquid (BTL) plant [62]. The objective is to maximize the flow rate of purified syngas,
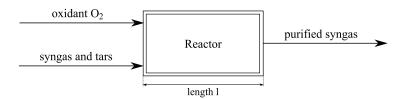

FIG. 7.2. *Schematic of the tar removal process in syngas production. Tar-polluted syngas enters the reactor from the left together with oxygen. Chemical reactions to reduce the tars take place in the reactor, and purified syngas exits the process at right.*

$\mathcal{F}_s$, at the outlet of the reactor by removing tar from the inlet stream as much as possible. The design parameters for the process are the length, $x_1 = l$, of the reactor and the inflow rate of oxygen, $x_2 = \mathcal{F}_{O_2}$. A call to the tar removal process simulator yields the outputs

$$\left(\mathcal{F}_s(x_1, x_2), T_{out}(x_1, x_2) \right),$$

where T_{out} is the temperature of the purified syngas at the outlet. We remark that the removal of tar is implicitly achieved by maximizing $\mathcal{F}_s$ and is therefore not explicitly included in the objective function. The set of feasible design parameters is restricted by physical and economical constraints, as well as safety standards for the operation of the BTL plant. On the one hand, the length of the reactor has to be sufficiently large to contain the syngas and allow it to react. On the other hand, building too large of a reactor would result in unallowable material costs. These considerations yield the restrictions $0.5\,[\text{m}] \leq x_1 \leq 2\,[\text{m}]$ for the extent of the reactor. The flow rate of the oxygen at the reactor inlet must also obey constraints. A lower bound must be respected in order to sustain the reformer process, while an upper bound is again dictated from an economical perspective, to limit operational costs. We thus impose the constraints $1334375\,[\text{kmol/h}] \leq x_2 \leq 3125000\,[\text{kmol/h}]$. Finally, because the reactor vessel might fail when the outlet temperature T_{out} exceeds a limit of 1680 Kelvin, safety concerns impose the constraint on the temperature $T_{out} \leq 1680\,[\text{K}]$. In summary, the feasible domain $X = \{(x_1, x_2)^T \in \mathbb{R}^2 \; : \; c(x_1, x_2) \leq 0\}$ is given by the constraints

$$c(x_1, x_2) = \begin{pmatrix} T_{out}(x_1, x_2) - 1680 \\ 0.5 - x_1 \\ x_1 - 2 \\ 1334375 - x_2 \\ x_2 - 3125000 \end{pmatrix}.$$

We note that the constraints on the length of the reactor and the flow rate of the oxygen are simple box constraints. However, the constraint on the temperature is more involved since T_{out} is only given by black-box evaluations of the tar removal process simulation. Overall, the constrained optimization problem can be stated as

$$\max_{c(x_1, x_2) \leq 0} \mathcal{F}_s(x_1, x_2). \tag{7.3}$$

We choose the starting point $x = (1.0, 1.5 \cdot 10^6)^T$ and rescale the second design parameter by 10^{-6} so that both design parameters are of the same order of magnitude. This scaling reduces

24

TABLE 7.6
Comparison of performance of the optimizers NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC on test problems in the Hock and Schittkowski benchmark set. TP denotes the number of the test problem, n and r the number of design variables and nonlinear constraints. The absolute error in the optimal design, and the absolute and relative errors in the optimal objective values are shown in the columns d_x, d_f^{abs}, and d_f^{rel} respectively. For all solvers a stopping threshold of 10^{-3} is used.

TP	n	r	SOLVER	#eval	d_x	d_f^{abs}	d_f^{rel}
29	3	1	NOWPAC	50	$1.0441 \cdot 10^{-4}$	$3.2215 \cdot 10^{-7}$	$1.4237 \cdot 10^{-8}$
			COBYLA	73	$7.6908 \cdot 10^{-4}$	$5.5365 \cdot 10^{-7}$	$2.4468 \cdot 10^{-8}$
			NOMAD	277	$2.9345 \cdot 10^{-3}$	$4.0290 \cdot 10^{-5}$	$1.7806 \cdot 10^{-6}$
			SDPEN	115	1.3449	5.8696	$2.5940 \cdot 10^{-1}$
			GSS-NLC	221	$2.9625 \cdot 10^{-1}$	$1.6742 \cdot 10^{-1}$	$7.3989 \cdot 10^{-3}$
43	4	3	NOWPAC	66	$9.8067 \cdot 10^{-6}$	$4.4098 \cdot 10^{-9}$	$1.0022 \cdot 10^{-10}$
			COBYLA	79	$1.4830 \cdot 10^{-3}$	$1.2760 \cdot 10^{-6}$	$2.9000 \cdot 10^{-8}$
			NOMAD	234	0	0	0
			SDPEN	174	1.8322	$2.1379 \cdot 10^{1}$	$4.8588 \cdot 10^{-1}$
			GSS-NLC	1111	$1.0311 \cdot 10^{-1}$	$5.0000 \cdot 10^{-2}$	$1.1364 \cdot 10^{-3}$
100	7	4	NOWPAC	155	$1.0563 \cdot 10^{-2}$	$2.8696 \cdot 10^{-4}$	$4.2161 \cdot 10^{-7}$
			COBYLA	237	$1.3533 \cdot 10^{-2}$	$1.1572 \cdot 10^{-3}$	$1.7002 \cdot 10^{-6}$
			NOMAD	804	$2.5400 \cdot 10^{-1}$	$2.5474 \cdot 10^{-1}$	$3.7427 \cdot 10^{-4}$
			SDPEN	251	$8.3322 \cdot 10^{-1}$	4.1820	$6.1444 \cdot 10^{-3}$
			GSS-NLC	1269	$7.6222 \cdot 10^{-1}$	3.5699	$5.2451 \cdot 10^{-3}$
113	10	8	NOWPAC	141	$7.9201 \cdot 10^{-4}$	$3.2979 \cdot 10^{-6}$	$1.3568 \cdot 10^{-7}$
			COBYLA	335	$9.5021 \cdot 10^{-3}$	$1.8091 \cdot 10^{-4}$	$7.4429 \cdot 10^{-6}$
			NOMAD	1199	1.7572	5.5933	$2.3012 \cdot 10^{-1}$
			SDPEN	392	1.1885	4.0313	$1.6586 \cdot 10^{-1}$
			GSS-NLC	4193	$8.1851 \cdot 10^{-1}$	2.1238	$8.7376 \cdot 10^{-2}$
227	2	2	NOWPAC	18	$6.5285 \cdot 10^{-6}$	$8.3104 \cdot 10^{-6}$	$8.3104 \cdot 10^{-6}$
			COBYLA	18	$5.8463 \cdot 10^{-5}$	$1.0860 \cdot 10^{-4}$	$1.0860 \cdot 10^{-4}$
			NOMAD	102	0	0	0
			SDPEN	90	$4.0569 \cdot 10^{-3}$	$5.7538 \cdot 10^{-3}$	$5.7538 \cdot 10^{-3}$
			GSS-NLC	393	0	0	0
228	2	2	NOWPAC	28	$9.8407 \cdot 10^{-5}$	$1.1405 \cdot 10^{-8}$	$3.8017 \cdot 10^{-9}$
			COBYLA	48	$1.9082 \cdot 10^{-3}$	$3.7562 \cdot 10^{-6}$	$1.2521 \cdot 10^{-6}$
			NOMAD	101	0	0	0
			SDPEN	4	3.0000	3.0000	1.0000
			GSS-NLC	287	0	0	0
264	4	3	NOWPAC	53	$1.6402 \cdot 10^{-4}$	$3.0458 \cdot 10^{-7}$	$6.9222 \cdot 10^{-9}$
			COBYLA	77	$1.3421 \cdot 10^{-3}$	$1.5297 \cdot 10^{-6}$	$3.4765 \cdot 10^{-8}$
			NOMAD	233	0	0	0
			SDPEN	174	1.8322	$2.1379 \cdot 10^{1}$	$4.8588 \cdot 10^{-1}$
			GSS-NLC	1352	0	0	0
285	15	10	NOWPAC	177	$7.8426 \cdot 10^{-5}$	$1.0413 \cdot 10^{-5}$	$1.2618 \cdot 10^{-9}$
			COBYLA	362	$1.5656 \cdot 10^{-3}$	$3.3056 \cdot 10^{-4}$	$4.0058 \cdot 10^{-8}$
			NOMAD	787	$3.3862 \cdot 10^{-1}$	$3.6380 \cdot 10^{1}$	$4.4087 \cdot 10^{-3}$
			SDPEN	650	3.1240	$3.7707 \cdot 10^{3}$	$4.5695 \cdot 10^{-1}$
			GSS-NLC	4647	1.2390	$5.9800 \cdot 10^{2}$	$7.2467 \cdot 10^{-2}$

TABLE 7.7
Comparison of performance of the optimizers NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC on test problems in the Hock and Schittkowski benchmark set. TP denotes the number of the test problem, n and r the number of design variables and nonlinear constraints. The absolute error in the optimal design and the absolute and relative errors in the optimal objective values are shown in the columns d_x, d_f^{abs}, and d_f^{rel} respectively. For all solvers a stopping threshold of 10^{-5} is used.

TP	n	r	SOLVER	#eval	d_x	d_f^{abs}	d_f^{rel}
29	3	1	NOWPAC	58	$1.2405 \cdot 10^{-5}$	$3.9934 \cdot 10^{-10}$	$1.7648 \cdot 10^{-11}$
			COBYLA	117	$8.7405 \cdot 10^{-6}$	$1.9876 \cdot 10^{-10}$	$8.7839 \cdot 10^{-12}$
			NOMAD	623	$2.8503 \cdot 10^{-4}$	$1.8000 \cdot 10^{-7}$	$7.9550 \cdot 10^{-9}$
			SDPEN	154	1.3450	5.8661	$2.5925 \cdot 10^{-1}$
			GSS-NLC	696	$2.9542 \cdot 10^{-1}$	$1.5742 \cdot 10^{-1}$	$6.9569 \cdot 10^{-3}$
43	4	3	NOWPAC	74	$9.8067 \cdot 10^{-6}$	$4.4098 \cdot 10^{-9}$	$1.0022 \cdot 10^{-10}$
			COBYLA	128	$5.2627 \cdot 10^{-6}$	$1.2718 \cdot 10^{-9}$	$9.1647 \cdot 10^{-10}$
			NOMAD	330	0	0	0
			SDPEN	228	1.8317	$2.1367 \cdot 10^1$	$4.8562 \cdot 10^{-1}$
			GSS-NLC	2332	$1.0232 \cdot 10^{-1}$	$5.0000 \cdot 10^{-2}$	$1.1364 \cdot 10^{-3}$
100	7	4	NOWPAC	238	$6.7890 \cdot 10^{-4}$	$1.2721 \cdot 10^{-6}$	$1.8690 \cdot 10^{-9}$
			COBYLA	729	$5.0509 \cdot 10^{-4}$	$6.2378 \cdot 10^{-7}$	$9.1647 \cdot 10^{-10}$
			NOMAD	2606	$2.2263 \cdot 10^{-1}$	$1.2558 \cdot 10^{-1}$	$1.8450 \cdot 10^{-4}$
			SDPEN	360	$8.3304 \cdot 10^{-1}$	4.1351	$6.0753 \cdot 10^{-3}$
			GSS-NLC	2769	$7.6247 \cdot 10^{-1}$	3.5699	$5.2451 \cdot 10^{-3}$
113	10	8	NOWPAC	188	$1.6343 \cdot 10^{-4}$	$3.4602 \cdot 10^{-8}$	$1.4236 \cdot 10^{-9}$
			COBYLA	635	$1.1470 \cdot 10^{-4}$	$2.3968 \cdot 10^{-8}$	$9.8609 \cdot 10^{-10}$
			NOMAD	1715	1.7512	5.5030	$2.2640 \cdot 10^{-1}$
			SDPEN	531	1.1885	4.0045	$1.6475 \cdot 10^{-1}$
			GSS-NLC	7172	$8.2086 \cdot 10^{-1}$	2.1338	$8.7788 \cdot 10^{-2}$
227	2	2	NOWPAC	31	$9.1139 \cdot 10^{-12}$	$1.2971 \cdot 10^{-11}$	$1.2971 \cdot 10^{-11}$
			COBYLA	26	$7.0430 \cdot 10^{-9}$	$1.1950 \cdot 10^{-8}$	$1.1950 \cdot 10^{-8}$
			NOMAD	158	0	0	0
			SDPEN	130	$2.2254 \cdot 10^{-5}$	$3.1472 \cdot 10^{-5}$	$3.1472 \cdot 10^{-5}$
			GSS-NLC	930	0	0	0
228	2	2	NOWPAC	31	$9.8407 \cdot 10^{-5}$	$1.1405 \cdot 10^{-8}$	$3.8017 \cdot 10^{-9}$
			COBYLA	67	$1.2070 \cdot 10^{-5}$	$1.3663 \cdot 10^{-10}$	$4.5543 \cdot 10^{-11}$
			NOMAD	174	0	0	0
			SDPEN	84	0	0	0
			GSS-NLC	779	0	0	0
264	4	3	NOWPAC	63	$5.9984 \cdot 10^{-6}$	$2.4949 \cdot 10^{-10}$	$5.6703 \cdot 10^{-12}$
			COBYLA	135	$1.3461 \cdot 10^{-5}$	$1.4405 \cdot 10^{-10}$	$3.2738 \cdot 10^{-12}$
			NOMAD	349	0	0	0
			SDPEN	228	1.8317	$2.1367 \cdot 10^1$	$4.8562 \cdot 10^{-1}$
			GSS-NLC	2441	0	0	0
285	15	10	NOWPAC	209	$1.9381 \cdot 10^{-5}$	$3.5764 \cdot 10^{-7}$	$4.3339 \cdot 10^{-11}$
			COBYLA	614	$1.6457 \cdot 10^{-5}$	$2.4447 \cdot 10^{-8}$	$2.9626 \cdot 10^{-12}$
			NOMAD	1246	$3.3808 \cdot 10^{-1}$	$3.5863 \cdot 10^1$	$4.3460 \cdot 10^{-3}$
			SDPEN	862	3.1240	$3.7707 \cdot 10^3$	$4.5694 \cdot 10^{-1}$
			GSS-NLC	9745	1.2307	$5.9200 \cdot 10^2$	$7.1740 \cdot 10^{-2}$

TABLE 7.8
Summarized performance statistics of NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC in finding an optimal design for the tar removal process model. SC indicates the stopping criteria, which is the ρ_{min} threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA, NOMAD, SDPEN, and GSS-NLC. For NOWPAC, in parentheses behind the number of model evaluations, we report the number of evaluations saved due to early termination by the error indicator.

	SC	#eval	x^*	f^*
NOWPAC	10^{-3}	19(15)	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
COBYLA	10^{-3}	24	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7953 \cdot 10^3$
NOMAD	10^{-3}	212	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
SDPEN	10^{-3}	53	$(2.0000, 2.2188 \cdot 10^6)^T$	$3.7941 \cdot 10^3$
GSS-NLC	10^{-3}	102	$(2.0000, 2.2198 \cdot 10^6)^T$	$3.7956 \cdot 10^3$
NOWPAC	10^{-4}	19(22)	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
COBYLA	10^{-4}	26	$(2.0000, 2.2201 \cdot 10^6)^T$	$3.7961 \cdot 10^3$
NOMAD	10^{-4}	232	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
SDPEN	10^{-4}	63	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7959 \cdot 10^3$
GSS-NLC	10^{-4}	197	$(2.0000, 2.2198 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
NOWPAC	10^{-5}	19(27)	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
COBYLA	10^{-5}	32	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
NOMAD	10^{-5}	266	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
SDPEN	10^{-5}	71	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$
GSS-NLC	10^{-5}	401	$(2.0000, 2.2199 \cdot 10^6)^T$	$3.7957 \cdot 10^3$

the anisotropy of the elliptical trust region and is tailored to the design space X. We summarize the performances of NOWPAC, COBYLA, NOMAD, SDPEN, and GSS-NLC in Table 7.8. We use NOWPAC's noise indicator and early termination feature in the solution of the tar removal problem. In Table 7.8 we report the number of function evaluations as well as the optimal designs and objective values at termination; we also state the number of saved function evaluations (for NOWPAC) in parentheses. NOMAD and GSS-NLC require considerably more objective function evaluations than NOWPAC, COBYLA, and SDPEN. We remark that COBYLA proposes designs that are not actually feasible, i.e., $T_{out}(x^*) - 1680 = 7.4 \cdot 10^{-3}$ and $T_{out}(x^*) - 1680 = 2.0 \cdot 10^{-5}$ for the stopping criteria $SC = 10^{-3}$ and $SC = 10^{-5}$ respectively. Constraint violations like this are, from a practical perspective, not desirable since they may cause failure of the whole reactor. We remark that NOMAD always proposes a feasible optimal design, whereas this is not guaranteed in SDPEN and GSS-NLC. In this example, however, both of the latter codes yield feasible solutions.

The tar removal process model is only available as a black-box simulator that exhibits irreducible numerical errors in its evaluations. Thus, as we have seen in the previous test examples, detection of inexact objective and constraint evaluations is necessary to prevent NOWPAC from performing superfluous iterations. We illustrate the errors of the black-box simulations in Figure 7.3 by plotting the output $(\mathcal{F}_S, T_{out})$ for various values of the oxygen inflow rates $\mathcal{F}_{O_2} \in [2.2198, 2.2200]$ and lengths of the reactor $l \in \{1.98, 1.99, 2.00\}$. To reveal the errors, we perform a transformation of the outputs; specifically, we subtract the approximated affine part of the solution. In Figure 7.4, we plot the norms of the Hessians of the objective function model and the first constraint model for a stopping criterion of $\rho_{min} = 10^{-5}$. We see that the error indicator marks iterations as non-convergent (with circles) as soon as the Hessian norms rise due to the inexact function evaluations. This shows the effectiveness of the error indicator.

8. Conclusions. This paper has presented a derivative-free trust region method for constrained nonlinear optimization. The method generalizes the work of Conn *et. al* [16, 20] to handle general black-box constraints without the need for derivative information. We provide a rigorous proof of
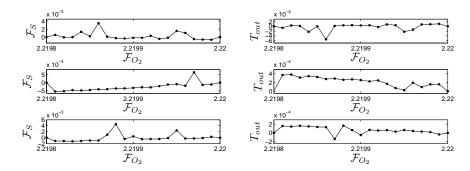
FIG. 7.3. *Affinely-transformed evaluations of the flowrate of syngas $\mathcal{F}_S$ (left) and reactor temperature T_{out} (right) with varying oxygen inflow $\mathcal{F}_{O_2} \in [2.2198, 2.2200]$ at reactor lengths of $l = 1.98$ (top), $l = 1.99$ (middle), $l = 2.00$ (bottom).*
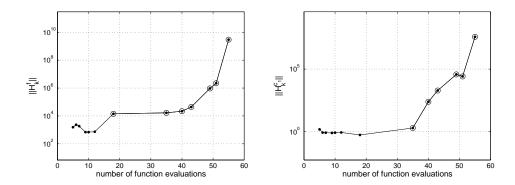

FIG. 7.4. *Norms of the Hessians of the models of the tar removal process for the objective (left) and the temperature constraint (right). Circles indicate the non-convergent regime.*

convergence of the method to first-order critical points. This result, given in Section 4, assumes that evaluations of the objective function and constraints are sufficiently accurate for the trust region surrogate models to satisfy the conditions in (2.1). In many practical applications, however, where evaluations of the objective function and the constraints are obtained via calls to a black-box simulation, we may only have access to inexact evaluations whose accuracy cannot be tuned. These inaccuracies in the evaluations may corrupt the full linearity properties (2.1) of the models m_k^f and $\{m_k^{c_i}\}_{i=1}^r$. In Section 6, we therefore derive an asymptotic bound on the decay rate of the errors, with respect to the trust region radii, to guarantee convergence. This theoretical analysis leads to the introduction of a error indicator τ, based on the norms of successive model Hessians. The indicator can be used to terminate NOWPAC iterations once they enter a regime where level of inaccuracy in the evaluations of the objective function and constraints is expected to impede further progress of the algorithm.

We note that the termination approach proposed in this paper is local and perhaps conservative. One might encounter situations in which the objective function or the constraints possess regions where the possible local descent is of the order of the accuracy level of the evaluations, yet after passing through this region, significant descent again becomes possible. In these situations, a less conservative behavior of the error indicator would be desirable. We therefore suggest the option of terminating NOWPAC only after a user-prescribed number of non-convergent iterations, essentially to allow for local randomized exploration in flat areas of the objective function.

Since NOWPAC is solely based on evaluations of the objective function and the constraints, it is applicable to a broad class of optimization problems for which no derivative information is available.

28

Moreover, it is guaranteed to converge to first-order critical points without incurring errors due to approximations of the feasible domain. We emphasize that Algorithm 1 is a skeleton procedure, wherein the user can chose the most appropriate methods for the computation of the trial step and the most suitable approximation method for the surrogates m_k^f and $\{m_k^{c_i}\}_{i=1}^r$.

In future work, we will explore application of the NOWPAC framework to problems in stochastic optimization. As we discussed briefly in Section 6, in stochastic programming the objective function and/or constraints are often replaced by averages or other measures of variability or risk associated with a lack of knowledge (see, e.g., [10, 56]). These quantities are often estimated using sampling strategies that exhibit uncorrelated errors between neighboring designs, and these errors are thus perfectly detectable by our error indicator τ. Our future work may therefore extend the concept of the error indicator into a feedback scheme for adaptively adjusting the accuracy of the objective/constraint evaluations (e.g., choosing the number of samples in a Monte Carlo approximation) to save computational costs while still guaranteeing convergence.

Appendix A. Auxiliary lemmas. In this appendix we provide auxiliary results that justify Assumptions 2.1 and 3.1, in order to complete the proofs in the main sections of this paper. First, Lemma A.1 shows that minimum-Frobenius-norm models satisfy Assumption 2.1. Lemma A.2 shows that local convexification of the feasible domain is possible by choosing a sufficiently large inner boundary path constant $\varepsilon_b > 0$. This results allows us, in Lemma A.3, to prove continuity of the criticality measures used in Lemma 4.2. Lemma A.4 establishes a relationship between the approximated criticality measure $\alpha_k(\rho_k)$ and its extended version $\mathfrak{A}_2[x_k]$. Thereafter we state, for use in Theorem 4.8, Corollary A.5, which is a direct consequence of statements in the proof of Lemma 4.2. Finally we present Lemma A.6, which complements the proof of Theorem 6.1.

LEMMA A.1 (Existence of a bounding function satisfying Assumption 2.1 for minimum-Frobenius-norm models). *Let m_x^c be a quadratic minimum-Frobenius-norm model of the constraint c. A bounding function satisfying Assumption 2.1 is given by*

$$b_c(s; x, \rho) = \|s\| \kappa_{dc} \rho + \begin{cases} \left(\frac{1}{2} \kappa_{bh} + \nu^c \right) \|s\|^2 & for & \|s\| \leq \rho, \\ \bar{b}_c(s; x, \rho) & else \end{cases}$$

with

$$\bar{b}_c(s; x, \rho) := \left(\kappa_{dc} + \frac{1}{2} \kappa_{bh} + \nu^c \right) \rho^2 P_0(\|s\|) + \left(\kappa_{dc} + 2 \left(\nu^c + \frac{1}{2} \kappa_{bh} \right) \right) \rho(1 - \rho) P_1(\|s\|) +$$

$$\left(\left(\kappa_{dc} + \frac{1}{2} \kappa_{bh} + \nu^c \right) \rho^2 + \bar{\kappa}_{\lambda_1} (1 - \rho) \rho \right) P_3(\|s\|) +$$

$$\left(\kappa_{dc} + 2 \left(\nu^c + \frac{1}{2} \kappa_{bh} \right) \right) \rho^2 (1 - \rho) P_4(\|s\|),$$

where $\bar{\kappa}_{\lambda_1} > 0$, κ_{dc} is the constant from the fully linear property (2.1d), and ν^c denotes the Lipschitz constant of the gradient of c. The functions $P_0, \ldots, P_4$ denote the cubic spline interpolation basis.

Proof. Let $x_k \in X$ and $\rho_k \in]0, \rho_{max}]$ be arbitrary and consider the approximation error $\varepsilon_c(x_k + s) := c(x_k + s) - m_k^c(x_k + s)$. Adding $c(x_k) - m_k^c(x_k) = 0$ (note that m_k^c interpolates c at

the points x_k) results in

$$\varepsilon_c(x_k + s) + c(x_k) - c(x_k + s) = m_k^c(x_k) - m_k^c(x_k + s) = -s^T g_k^c - \frac{1}{2} s^T H_k^c s$$

$$= -s^T (g_k^c + H_k^c s) + \frac{1}{2} s^T H_k^c s = -s^T \nabla m_k^c(x_k + s) + \frac{1}{2} s^T H_k^c s.$$

A Taylor expansion of $c(x_k)$ around $x_k + s$,

$$c(x_k) = c(x_k + s - s) = c(x_k + s) - s^T \nabla c(x_k + s) - s^T (\nabla c(x_k + \theta s) - \nabla c(x_k + s)),$$

for some $\theta \in]0, 1[$, yields

$$\varepsilon_c(x_k + s) - s^T \nabla c(x_k + s) - s^T (\nabla c(x_k + \theta s) - \nabla c(x_k + s)) = -s^T \nabla m_k^c(x_k + s) + \frac{1}{2} s^T H_k^c s,$$

i.e.,

$$\varepsilon_c(x_k + s) = -s^T \nabla m_k^c(x_k + s) + \frac{1}{2} s^T H_k^c s + s^T \nabla c(x_k + s) + s^T (\nabla c(x_k + \theta s) - \nabla c(x_k + s))$$

$$= s^T (\nabla c(x_k + s) - \nabla m_k^c(x_k + s)) + \frac{1}{2} s^T H_k^c s + s^T (\nabla c(x_k + \theta s) - \nabla c(x_k + s)).$$

It follows that

$$|\varepsilon_c(x_k + s)| \leq \|s\| \kappa_{dc} \rho_k + \left(\frac{1}{2} \kappa_{bh} + \nu^c (1 - \theta) \right) \|s\|^2,$$

where ν^c is the Lipschitz constant of the gradient of c. Since (x_k, ρ_k) was chosen arbitrarily in $X \times [0, \rho_{max}]$, the bounding function restricted to $B(0, \rho)$ is given by

$$b_c(s; x, \rho) = \|s\| \kappa_{dc} \rho + \left(\frac{1}{2} \kappa_{bh} + \nu^c \right) \|s\|^2.$$

We see that b_c is convex and radial symmetric in s, as well as constant in x. If $\rho < 1$, we use polynomial Hermite interpolation to smoothly extend the bounding function to $B(0, 1)$ such that b_c is continuous in $(s; x, \rho) \in B(0, 1) \times X \times [0, 1]$. $\square$

LEMMA A.2. *Let $c : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient on $\mathcal{L} \subset \mathbb{R}^n$. For every $x \in \mathcal{L}$ with $c(x) \leq 0$ and $\rho > 0$ there exists an $\varepsilon_b > 0$ such that $X_{(x,\rho)}^-$, $X_{(x,\rho)}^+$, along with X_x^{ibp} and $X_x \cap B(x, 1)$ for all $(x, \rho) \in X \times [0, \rho_{max}]$, are strictly convex sets.*

Proof. We first show the existence of a large enough $\varepsilon_{b,1} > 0$ such that X_x^{ibp} is strictly convex. The first-order Taylor approximation of the inner boundary path $h_x(\eta)$, $\eta \in B(x, 1)$ around $\xi \in B(x, 1)$ results in

$$h_x(\eta) - h_x(\xi) = \langle \nabla h_x(\xi), \eta - \xi \rangle + \varepsilon_{b,1} R(\xi, \eta)$$

with non-negative $R(\xi, \eta) \in \Omega(\|\eta - \xi\|^2)$. Moreover, due to the constraints c being continuously differentiable with Lipschitz continuous gradient, there exists a function $T(\xi, \eta)$ such that

$$c(\eta) - c(\xi) = \langle \nabla c(\xi), \eta - \xi \rangle + T(\xi, \eta)$$

with $T(\xi, \eta) \in \mathcal{O}(\|\eta - \xi\|^2)$. Thus we obtain

$$c(\eta) + h_x(\eta) - c(\xi) + h_x(\xi) = \langle \nabla c(\xi) + \nabla h_x(\xi), \eta - \xi \rangle + T(\xi, \eta) + \varepsilon_{b,1} R(\xi, \eta).$$

Now, choosing $\varepsilon_{b,1} > 0$ such that $T(\xi, \eta) + \varepsilon_{b,1} R(\xi, \eta) > 0$ results in

$$c(\eta) + h_x(\eta) - c(\xi) + h_x(\xi) > \langle \nabla c(\xi) + \nabla h_x(\xi), \eta - \xi \rangle,$$

which shows the strict convexity of X_x^{ibp}. The existence of an $\varepsilon_{b,2} > 0$ such that $X_x \cap B(x_x, 1)$ is strictly convex can be shown analogously by replacing the constraints c with M_x^c, i.e., the extended surrogate models (3.3), which have the same smoothness properties as the constraints c themselves. Moreover, since the bounding function b_c is convex, replacing c by $c + b_c$ in the above proof yields the existence of a constant $\varepsilon_{b,+} > 0$ such that $X_{(x,\rho)}^+$ is strictly convex. For the same choice, $\varepsilon_{b,-} = \varepsilon_{b,+}$, the outer approximation $X_{(x,\rho)}^-$ is also strictly convex. This immediately follows by noting that the offset $\kappa_{\lambda_1} \rho$ to the constraints in $X_{(x,\rho)}^-$ is constant in x, and thus does not affect the strict convexity of the set. The assertion of the lemma now follows with $\varepsilon_b := \max\{\varepsilon_{b,1}, \varepsilon_{b,2}, \varepsilon_{b,+}\}$. $\square$

LEMMA A.3. *The set X_x^{ibp} as well as the inner and outer approximations $X_{(x,\rho)}^{\pm}$ are upper and lower semi-continuous in (x, ρ) on the domain $X \times [0, \rho_{max}]$. Moreover, the criticality measure $\mathfrak{A}[x]$ as well as the lower and upper bounds $\mathfrak{A}_1^{\pm}[x, \rho]$ are continuous on X and $X \times [0, \rho_{max}]$ respectively.*
Proof. First we discuss the continuity of

$$\mathfrak{A}_1^+[x, \rho] = - \min_{\xi \in X_{(x,\rho)}^+} \langle \nabla f(x), \xi - x \rangle$$

in $X \times [0, \rho_{max}]$. For this we use [23, Thm. 2.1] and remark that the objective $- \langle \nabla f(x), \xi - x \rangle$ is continuous in (ξ, x, ρ). Note that we included ρ for completeness; since the objective is constant in ρ, it is trivially continuous with respect to this variable. Thus, we have to show upper and lower semi-continuity of the feasible set

$$X_{(x,\rho)}^+ = \{x + d \ : \ c(x + d) + h_x(x + d) + b_c(d; x, \rho) \leq 0\} \cap B(x, 1).$$

Note that $X_{(x,\rho)}^+ \subseteq B(x, 1)$. Let $N(\bar{x}, \bar{\rho})$ be a bounded neighborhood of $(\bar{x}, \bar{\rho}) \in X \times [0, \rho_{max}]$. Then there exists a finite bound $d_x > 0$ such that $\|x_1 - x_2\| \leq d_{\bar{x}}$ for all $x_1, x_2 \in N(\bar{x}, \bar{\rho})$. It therefore holds that

$$\bigcup_{(x,\rho) \in N(\bar{x}, \bar{\rho})} X_{(x,\rho)}^+ \subseteq B(\bar{x}, 1 + d_{\bar{x}}),$$

i.e., X^+ is uniformly compact; see [23, p. 217]. Moreover, since $\mathcal{L}$ is compact (see Assumption 2.2(a)) there exists a ball $B(0, d_{\mathcal{L}}) \subset \mathcal{L}$ such that

$$X_{(\bar{x}, \bar{\rho})}^+ = \{\xi \in B(0, d_{\mathcal{L}}) \ : \ g(\xi; x, \rho) \leq 0, \ i = 0 \ldots r\},$$

with the strictly convex continuous functions $g_0(\xi; x, \rho) := \|\xi - x\|^2 - 1$ and $g_i(\xi; x, \rho) := c(\xi) + b_c(\xi - x; x, \rho) + h_x(\xi)$, $i = 1 \ldots r$; see Assumption 3.1. It now follows from [23, Thm. 2.9] that X^+ is open and closed at every $(x, \rho) \in X \times [0, \rho_{max}]$, which in turn implies lower and upper semi-continuity of X^+ relative to $X \times [0, \rho_{max}]$; see [23, p. 217]. Finally, [23, Thm. 2.1] yields the continuity of $\mathfrak{A}_1^+[x, \rho]$ in $X \times [0, \rho_{max}]$.
The continuity of $\mathfrak{A}^-[x, \rho]$ and $\mathfrak{A}[x]$ follows analogously by considering the functions $g_i(\xi; x, \rho) := c(\xi) - \kappa_{\lambda_3} \rho + h_x(\xi)$ and $g_i(\xi; x, \rho) := c(\xi) + h_x(\xi)$ for $i = 1 \ldots r$, respectively. $\square$

LEMMA A.4. *If X_k is convex and $x_k \in X_k$, with the corresponding trust region radius ρ_k defined by Algorithm 1, then $\mathfrak{A}_2[x_k] \leq \alpha_k(\rho_k)$ for all $0 < \rho_k < 1$.*
Proof. Let us recall the definitions of the intermediate and approximated criticality measures

$$\mathfrak{A}_2[x_k] = \left| \min_{\substack{x_k + s \in X_k \\ \|s\| \leq 1}} \left\langle g_k^f, s \right\rangle \right| \quad \text{and} \quad \alpha_k(\rho_k) = \frac{1}{\rho_k} \left| \min_{\substack{x_k + d \in X_k \\ \|d\| \leq \rho_k}} \left\langle g_k^f, d \right\rangle \right|.$$

First we note that in the trivial case of $g_k^f = 0$, $\mathfrak{A}_2[x_k] = \alpha_k(\rho_k) = 0$. From here on, we therefore assume that $g_k^f \neq 0$. We denote the optimal solutions of the criticality subproblems by

$$\hat{s}_k := \arg\min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \left\langle g_k^f, s \right\rangle \quad \text{and} \quad \hat{d}_k := \arg\min_{\substack{x_k+d\in X_k \\ \|d\|\leq \rho_k}} \left\langle g_k^f, d \right\rangle.$$

Also, let $\mathcal{N}_k$ denote the normal cone to all constraints $m_k^c + h_k$ that are active in $x_k + \hat{d}_k$. We proceed by looking at the three possible positions of the optimal solution $x_k + \hat{d}_k$ individually. Case 1 addresses situations where $x_k + \hat{d}_k$ lies on the trust region boundary, but not on the boundary of X_k. Case 2 addresses situations where $x_k + \hat{d}_k$ lies on the boundary of X_k but not on the trust region boundary. Case 3 considers the situation where $x_k + \hat{d}_k$ is constrained both by the boundary of X_k and by the trust region boundary.

Case 1: $m_k^c(x_k + \hat{d}_k) + h_k(x_k + \hat{d}_k) < 0$:

In this case only the trust region constraint is active. Thus, $\|\hat{d}_k\| = \rho_k$ and $\hat{d}_k$ is aligned with $-g_k^f$, yielding

$$\alpha_k(\rho_k) = -\frac{1}{\rho_k} \left\langle g_k^f, \hat{d}_k \right\rangle = \frac{\left\langle g_k^f, g_k^f \right\rangle}{\|g_k^f\|} = \|g_k^f\|. \tag{A.1}$$

On the other hand we have

$$\mathfrak{A}_k[x_k] = -\min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \left\langle g_k^f, s \right\rangle \leq -\min_{x_k+s\in B(x_k,1)} \left\langle g_k^f, s \right\rangle = \|g_k^f\| = \alpha_k(\rho_k),$$

where we used (A.1) in the last equality.

Case 2: $\hat{d}_k \in \mathcal{N}_k$:

In this case $\hat{d}_k$ lies within the normal cone to the boundary of X_k. Moreover, relaxing the trust region constraint $\|d\| \leq \rho_k$ to $\|d\| \leq 1$ has no influence on $\hat{d}_k$, i.e., $\hat{s}_k = \hat{d}_k$. Thus, from the definition of $\mathfrak{A}_2[x_k]$ and the approximated criticality measure $\alpha_k(\rho_k)$, we have $\mathfrak{A}_2[x_k] = \rho_k \alpha_k(\rho_k) \leq \alpha_k(\rho_k)$.
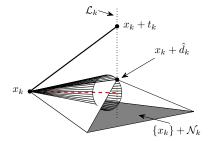
Case 3: $\hat{d}_k \notin \mathcal{N}_k$, $\mathcal{N}_k \neq \{0\}$:

First note that $\mathcal{N}_k \neq \{0\}$ implies that $x_k + \hat{d}_k$ does not lie within the interior of X_k. Moreover, since $\hat{d}_k \notin \mathcal{N}_k$, at least one constraint *and* the trust region constraint are active at $x_k + \hat{d}_k$. Thus $\|\hat{d}_k\| = \rho_k$ and $\rho_k^{-1}\hat{d}_k$ is a vector of unit length. Since $\hat{d}_k$ is an optimal solution of the criticality subproblem for α_k, the negative gradient $-g_k^f$ lies within the normal cone $\mathcal{C}_k$ to the overall set $X_k \cap B(x_k, \rho_k)$ at the point $x_k + \hat{d}_k$, i.e., it can be written as a convex combination of elements in $\mathcal{N}_k$ and $\hat{d}_k$, i.e., $\mathcal{C}_k = \text{conv}(\{\hat{d}_k\} \cup \mathcal{N}_k)$. In Figure A.1 the cone $\mathcal{C}_k$ is depicted as a pyramidal cone bounded by the gray cone $\mathcal{N}_k$ and the two white triangular faces. We decompose the gradient g_k^f into two components $g_k^f = g_k^\mathcal{N} + g_k^{\mathcal{N}\perp}$, where $g_k^\mathcal{N} \in \mathcal{N}_k$ and $g_k^{\mathcal{N}\perp}$ is orthogonal to $\mathcal{N}_k$. Note that $g_k^\mathcal{N} \neq 0$ and $g_k^{\mathcal{N}\perp} \neq 0$ since $\hat{d}_k \notin \mathcal{N}_k$ and $g_k^f \in \mathcal{C}_k$. We denote the set of indices of constraints that are active in $x_k + \hat{d}_k$ by

$$\mathcal{I}_k := \left\{ i \; : \; c_i\left(x_k + \hat{d}_k\right) = 0 \right\}.$$

Consider the linearized feasible domain $\mathcal{T}_k$ defined by the set of points where the linearizations of all constraints that are active in $x_k + \hat{d}_k$ are less or equal than zero. Due to the linearity of $\mathcal{T}_k$, the normal cone at any point on the boundary of $\mathcal{T}_k$ is equal to $\mathcal{N}_k$. Now set

$$t_k(\Delta) := \arg\min_{\substack{x_k+t\in\mathcal{T}_k \\ \|t\|\leq\Delta}} \left\langle g_k^f, t \right\rangle,$$
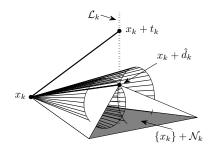
FIG. A.1. *Illustration of the (shifted) normal cone $\{x_k\} + \mathcal{N}_k$ (grey area, half hidden underneath the other two faces of $\mathcal{C}_k$) along with $\mathcal{L}_k$ (vertical dotted line). The dashed red line represents one vector $\nu \in \mathcal{C}_n$ and the circular cone $\mathcal{S}_k(\nu)$ surrounding it contains all vectors that enclose an angle with ν that is less than or equal to $\phi_{d,\nu}$. Right: two circular cones $\mathcal{S}_k(\nu_1)$ and $\mathcal{S}_k(\nu_1)$, $\nu_1, \nu_2 \in \mathcal{C}_n$.*

$\Delta \in [\rho_k, 1]$, and denote the line going through the points $x_k + \hat{d}_k$ and $x_k + t_k(1)$ by

$$\mathcal{L}_k := \left\{ x \in \mathbb{R}^n \ : \ x = x_k + \hat{d}_k + \sigma\left(t_k(1) - \hat{d}_k\right), \ \sigma \in \mathbb{R} \right\}.$$

Note that the line $\mathcal{L}_k$ is aligned with $g_k^{\mathcal{N}^\perp}$ and lies on the boundary of the linearized domain $\mathcal{T}_k$, since $g_k^{\mathcal{N}} \neq 0$. It holds that $t_k(\rho_k) = \hat{d}_k$, and the vector $t_k := t_k(1)$ is the solution of the approximated criticality subproblem subject to the linearized constraints at the point $x_k + \hat{d}_k$, intersected with $B(x_k, 1)$—i.e., subject to the constraint set $\mathcal{T}_k \cap B(x_k, 1)$. Since the line $\mathcal{L}_k$ is aligned with the gradient component $g_k^{\mathcal{N}^\perp}$, which in turn is orthogonal to the normal cone $\mathcal{N}_k$, the line $\mathcal{L}_k$ is itself orthogonal to $\mathcal{N}_k$.

Now we will show that the angle ϕ_t between t_k and $-g_k^f$ is greater than the angle ϕ_d between $\hat{d}_k$ and $-g_k^f$, i.e., $\phi_t \geq \phi_d$. More generally, we denote the angle between $\nu \in \mathcal{C}_n$ and $\hat{d}_k$ by $\phi_{d,\nu}$. For every vector $\nu \in \mathcal{C}_n$ (i.e., for every possible direction of $-g_k^f$) we consider the cylindrical cone $S_k(\nu)$ of all vectors in $\mathbb{R}^n$ that enclose an angle with ν equal to or smaller than $\phi_{d,\nu}$; see Figure A.1 (left) for an illustration. We define

$$\mathcal{U}_k := \bigcup_{\nu \in \mathcal{C}_n} \mathcal{S}_k(\nu).$$

as the set of all vectors v for which there exists a vector $\nu \in \mathcal{C}_n$ such that the angle between v and ν is smaller than the angle between $\hat{d}_k$ and ν, cf. Figure A.1 (right). Since $t_k \notin \mathcal{C}_n$ and $\mathcal{L}_k$ is orthogonal to $\mathcal{N}_k$ it follows that $t_k \notin \mathcal{U}_k$. In particular it holds for the special choice of $\nu = -g_k^f \in \mathcal{C}_n$ that $\phi_t > \phi_d$.

We now interpret the inner products in the criticality subproblems as cosines between the corresponding vectors, i.e.,

$$\rho_k \|g_k^f\| \cos(\phi_d) = -\left\langle g_k^f, \hat{d}_k \right\rangle \quad \text{and} \quad \|t_k\| \|g_k^f\| \cos(\phi_t) = -\left\langle g_k^f, t_k \right\rangle,$$

and conclude that

$$\begin{aligned}
\alpha_k(\rho_k) = -\frac{1}{\rho_k}\left\langle g_k^f, \hat{d}_k \right\rangle &= \frac{1}{\rho_k}\rho_k \|g_k^f\| \cos(\phi_d) \\
&\geq \|t_k\|\|g_k^f\| \cos(\phi_t) = -\min_{x_k + t \in \mathcal{T}(x_k + \hat{d}_k)} \left\langle g_k^f, t \right\rangle \geq \mathfrak{A}_2[x_k].
\end{aligned}$$

33

In the first inequality we used $\|t_k\| \leq 1$ and $\phi_t \geq \phi_d$, whereas in the last inequality we used the fact that X_k is convex, i.e., $X_k \subseteq \mathcal{T}_k$.

In summary, we see that in all cases the approximate criticality measure dominates $\mathfrak{A}_2[x_k]$, yielding the assertion of the lemma. $\square$

The assertion of the following corollary can be directly found in the proof of Lemma 4.2. However, for better readability, we formulate these results in a separate statement since they are also important for the proof of Theorem 4.8.

COROLLARY A.5. *Consider the sequence of points* $\{x_k\}_k \subset X$ *and the associated trust region radii* $\{\rho_k\}_k$ *with* $\rho_k \to 0$. *For every* $\varepsilon > 0$ *it holds that*

$$|\mathfrak{A}[x_k] - \mathfrak{A}_1[x_k]| < \varepsilon \quad and \quad |\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k]| < \varepsilon$$

for k *sufficiently large.*

Proof. Let us recall the definitions of the intermediate criticality measures $\mathfrak{A}_1$ and $\mathfrak{A}_2$ as given in Lemma 4.2,

$$\mathfrak{A}_1[x_k] = \left| \min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \langle \nabla f(x_k), s \rangle \right| \quad and \quad \mathfrak{A}_2[x_k] = \left| \min_{\substack{x_k+s\in X_k \\ \|s\|\leq 1}} \left\langle g_k^f, s \right\rangle \right|.$$

The first assertion of the corollary follows directly from the first part of the proof of Lemma 4.2 where we showed that $\lim_{k\to 0} |\mathfrak{A}[x_k] - \mathfrak{A}_1[x_k]| = 0$ if $\rho_k \to 0$. The second assertion directly follows from the second part of the proof of Lemma 4.2, where we showed that $\lim_{k\to 0} |\mathfrak{A}_1[x_k] - \mathfrak{A}_2[x_k]| = 0$ if $\rho_k \to 0$. $\square$

Finally we state Lemma A.6, which completes the proof of Theorem 6.1. It essentially follows directly from the proof of [20][Lemma 5.4]; however, we want to elaborate on the changes required to yield the particular assertion in which we are interested.

LEMMA A.6. *Under the assumptions of Theorem 6.1 it holds that*

$$\|\nabla f(x_k + s) - \nabla f_{n,k}(x_k + s)\| \in o(\rho_k) \quad and$$
$$\|\nabla c(x_k + s) - \nabla c_{n,k}(x_k + s)\| \in o(\rho_k).$$

Proof. First we define $y := x_k + s$ and recall that $\delta_k^{f_{n,k}}(y) = f_{n,k}(y) - f(y)$ and thus $\nabla \delta_k^{f_{n,k}}(y) = \nabla f_{n,k}(y) - \nabla f(y)$. Let us denote by $\{y_i\}$ the interpolation points on which we construct the minimum-Frobenius-norm model,

$$m_k^\delta(y) = c + g^T y + \frac{1}{2} y^T H y, \tag{A.2}$$

of $\delta_k^{f_{n,k}}(y)$. Now we interpret the minimum-Frobenius-norm model as a quadratic approximation of the constant zero function $\mu(y) \equiv 0$ based on inexact evaluations $\delta_k^{f_{n,k}}(y_i) \in o(\rho_k^2)$, i.e.,

$$m_k^\delta(y) = \mu(y) + e^\delta(y) \quad and \tag{A.3}$$
$$\nabla m_k^\delta(y) = g + Hy = \nabla \mu(y) + e^g(y). \tag{A.4}$$

Here, $e^\delta(y)$ and $e^g(y)$ are the approximation errors that result from the minimum-Frobenius-norm approximation as well as the inexact evaluations. Evaluating (A.2) at all interpolation points y_i and

subtracting (A.3) yields

$$\delta_k^{f_{n,k}}(y_i) - \mu(y) - e^\delta(y) = m_k^\delta(y_i) - m_k^\delta(y)$$
$$= g^T(y_i - y) + \frac{1}{2}(y_i - y)^T H(y_i - y) + (y_i - y)^T Hy$$
$$= (g + Hy)^T(y_i - y) + \frac{1}{2}(y_i - y)^T H(y_i - y)$$
$$= \nabla\mu(y)^T(y_i - y) + (e^g)^T(y_i - y) + \frac{1}{2}(y_i - y)^T H(y_i - y),$$

where we used (A.4) in the last equation. Now note that $\mu(y) = 0$ and $\nabla\mu(y) = 0$; thus

$$(e^g)^T(y_i - y) + \frac{1}{2}(y_i - y)^T H(y_i - y) = \delta_k^{f_{n,k}}(y_i) - e^\delta(y).$$

Subtracting the above equation for $i = 0$ from all the other equations for $i > 0$ yields

$$(e^g)^T(y_i - y_0) + \frac{1}{2}(y_i - y_0)^T H(y_i - y_0) = \delta_k^{f_{n,k}}(y_i) - \delta_k^{f_{n,k}}(y_0) = o(\rho_k^2),$$

which is exactly the type of equation discussed in the proof [20][Lemma 5.4]. The assertion of this lemma is therefore a straightforward consequence of [20][Lemma 5.4]. Replacing the objective function f with the constraints c, the same proof yields the second statement in the assertion of this lemma. □

REFERENCES

[1] M. A. ABRAMSON AND C. AUDET, *Convergence of mesh adaptive direct seach to second-order stationary points*, SIAM Journal on Optimization, 17 (2006), pp. 606–619.

[2] B. M. ADAMS, W. J. BOHNHOFF, K. R. DALBEY, J. P. EDDY, M. S. ELDRED, D. M. GAY, K. HASKELL, P. D. HOUGH, AND L. P. SWILER, *DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user's manual*, tech. report, Sandia Technical Report SAND2010-2183, 2013.

[3] A. AGARWAL AND L. T. BIEGLER, *A trust-region framework for constrained optimization using reduced order modeling*, Optimization and Engineering, 14 (2013), pp. 3–35.

[4] C. AUDET, A. L. CUSTODIO, AND J. E. DENNIS JR., *Erratum: mesh addaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 18 (2008), pp. 1501–1503.

[5] C. AUDET AND J. E. DENNIS JR., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 17 (2006), pp. 188–217.

[6] C. AUDET AND J. E. DENNIS JR., *A progressive barrier for derivative-free nonlinear programming*, SIAM Journal on Optimization, 20 (2009), pp. 445–472.

[7] M. S. BAZARAA, H. D. SHERALI, AND C. M. SHETTY, *Nonlinear programming: theory and algorithms*, Wiley & Sons, Inc., 3rd ed., 2006.

[8] D. P. BERTSEKAS, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Transactions on Automatic Control, 21 (1976), pp. 174–184.

[9] D. P. BERTSEKAS, *Nonlinear programming*, Athena Scientific, Bellmont, MA, 2nd edition ed., 1999.

[10] H. G. BEYER AND B. SENDHOFF, *Robust optimization - a comprehensive survey*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 3190–3218.

[11] S. BHATNAGAR, H. L. PRASAD, AND L. A. PRASHANTH, *Stochastic recursive algorithms for optimization*, vol. 434 of Lecture notes in control and information sciences, Springer-Verlag London Heidelberg New York Dordrecht, 2013.

[12] D. M. BORTZ AND C. T. KELLEY, *Computational methods for optimal design and control*, vol. 24 of Progress in Systems and Control Theory, de Gruyter, 1998, ch. The simplex gradient and noisy optimization problems, pp. 77–90.

[13] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2009.

[14] R. G. CARTER, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM Journal of Numerical Analysis, 28 (1991), pp. 251–265.

[15] T. D. CHOI AND C. T. KELLEY, *Superlinear convergence and implicit filtering*, SIAM Journal on Optimization, 10 (2000), pp. 1149–1162.

[16] A. R. Conn, N. Gould, A. Sartenaer, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM Journal on Optimization, 3 (1993), pp. 164–221.

[17] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-region methods*, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[18] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Geometry of interpolation sets in derivative free optimization*, Mathematical Programming, Series B, 111 (2008), pp. 141–172.

[19] ———, *Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points*, SIAM Journal on Optimization, 20 (2009), pp. 387–415.

[20] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, SIAM, Society for Industrial and Applied Mathematics and the Mathematical Programming Society, Philadelphia, 2009.

[21] F. E. Curtis, N. .I. M. Gould, D. P. Robinson, and P. L. Toint, *An interior-point trust-funnel algorithm for nonlinear optimization*, tech. report, Lehigh University, Lehigh Industrial and Systems Engineering, COR@L Technical Report 13T-010, 2013.

[22] G. Di Pillo, S. Lucidi, and F. Rinaldi, *A derivative-free algorithm for constrained global optimization based on exact penalty functions*, Journal of Optimization Theory and Applications, Springer Science+Business Media New York (2013).

[23] A. V. Fiacco and Y. Ishizuka, *Sensitivity and stability analysis for nonlinear programming*, Annals of Operations Research, 27 (1990), pp. 215–236.

[24] D. E. Finkel and C. T. Kelley, *Convergence analysis of the DIRECT algorithm*, tech. report, Technical Report CRSC-TR04-28, North Carolina State University, Center for Research in Scientific Computation, July 2004.

[25] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM Journal on Optimization, 13 (2002), pp. 635–659.

[26] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2002), pp. 239–269.

[27] N. I. M. Gould and Ph. L. Toint, *Nonlinear programming without penalty function or a filter*, Mathematical Programming, 122 (2010), pp. 155–196.

[28] G. A. Gray and T. G. Kolda, *Algorithm 856: APPSPACK 4.0: asynchronous parallel pattern search for derivative-free optimization*, ACM Transactions on Mathematical Software, 32 (2006), pp. 485–507.

[29] M. Heinkenschloss and L. N. Vicente, *Analysis of inexact trust-region SQP algorithms*, SIAM Journal on Optimization, 12 (2002), pp. 283–302.

[30] W. Hock and K. Schittkowski, *Lecture Notes in Economics and Mathematical Systems*, Springer, 1981, ch. Test examples for nonlinear programming, no. 187.

[31] R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, Journal of the ACM, 8 (1961), pp. 212–229.

[32] S. G. Johnson, *The NLopt nonlinear-optimization package*. http://ab-initio.mit.edu/nlopt.

[33] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, 79 (1993), pp. 157–181.

[34] A. Kannan and S. M. Wild, *Obtaining quadratic models of noisy functions*, Tech. Report ANL/MCS-P1975-1111, Argonne National Laboratory, 9700 South Cass Avenue Argonne, Illinois 60439, September 2012.

[35] C. T. Kelley, *Iterative methods for optimization*, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[36] H. J. Kushner and G. G. Yin, *Stochastic approximation algorithms and applications*, vol. 35 of Applications of mathematics, Springer Verlag New York, 1997.

[37] S. Le Digabel, *Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm*, ACM Transactions on Mathematical Software, 37 (2011), p. acticle no. 44.

[38] R. M. Lewis, V. Torczon, and M. W. Trosset, *Direct search methods: then and now*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 191–207.

[39] X. Liang, H. A. Bashir, and S. Li, *Applying infeasible interior point method to SQP for constrainted nonlinear programming*, International Conference on Computer Science and Software Engineering, 1 (2008), pp. 399–402.

[40] G. Liuzzi, S. Lucidi, and M. Sciandrone, *A derivative-free algorithm for linearly constrained finite minimax problems*, SIAM Journal on Optimization, 16 (2006), pp. 1054–1075.

[41] ———, *Sequential penalty derivative-free methods for nonlinear constrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2614–2635.

[42] A. March and K. Willcox, *Constrained multifidelity optimization using model calibration*, Structural and Multidisciplinary Optimization, 46 (2012), pp. 93–109.

[43] J. A. Nelder and R. Mead, *A simplex method for function minimization*, The Computer Journal, 7 (1965), pp. 308–313.

[44] T. D. Plantenga, *HOPSPACK 2.0 user manual*, Tech. Report SAND2009-6265, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, October 2009.

[45] M. J. D. Powell, *Computing methods in applied sciences and engineering*, vol. 704 of Lecture Notes in Math-

ematics, Springer-Verlag, 1977, ch. Variable metric methods for constrained optimization, pp. 62–72.

[46] M. J. D. POWELL, *Advances in Optimization and Numerical Analysis*, Kluwer Academic, Dordrecht, 1994, ch. A direct search optimization method that models the objective and constraint functions by linear interpolation, pp. 51–67.

[47] ———, *Direct search algorithms for optimization calculations*, Acta Numerica, 7 (1998), pp. 287–336.

[48] ———, *On the Lagrange functions of quadratic models that are defined by interpolation*, tech. report, DAMTP 2000/NA10, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 2000.

[49] ———, *UOBYQA: Unconstrained optimization by quadratic approximation*, Mathematical Programming, Series B, 92 (2002), pp. 555–582.

[50] ———, *On trust region methods for unconstrained minimization without derivatives*, Mathematical Programming, Series B, 97 (2003), pp. 605–623.

[51] ———, *Least Frobenius norm updating of quadratic models that satisfy interpolation conditions*, Mathematical Programming, Series B, 100 (2004), pp. 183–215.

[52] R. G. REGIS, *Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions*, Computers & Operations Research, 38 (2011), pp. 837–853.

[53] ———, *Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points*, Engineering Optimization, 46 (2014), pp. 218–243.

[54] L. M. RIOS AND N. V. SAHINIDIS, *Derivative-free optimization: A review of algorithms and comparison of software implementations*, Journal of Global Optimization, 56 (2013), pp. 1247–1293.

[55] H. ROBBINS AND S. MONRO, *A stochastic approximation method*, Annals of Mathematical Statistics, 22 (1951), pp. 400–407.

[56] R. T. ROCKAFELLAR, S. URYASEV, AND M. ZABARANKIN, *Deviation measures in risk analysis and optimization*, tech. report, Research Report 2002-7, Risk Management and Financial Engineering Lab, Center for Applied Optimization, University of Florida, 2002.

[57] P. R. SAMPAIO AND P. L. TOINT, *A derivative-free trust-funnel method for equality-constrained nonlinear optimization*, Compuational Optimization and Applications, 61 (2015), pp. 25–49.

[58] K. SCHITTKOWSKI, *More test examples for nonlinear programming codes*, in Lecture Notes in Economics and Mathematical Systems, Springer, 1987.

[59] ———, *306 test problems for nonlinear programming with optimal solutions - user's guide*, tech. report, University of Bayreuth, Department of Computer Science, 2008.

[60] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYNSKI, *Lectures on stochastic programming*, Society for Industrial and Applied Mathematics and the Mathematical Programming Society, 2009.

[61] W SPENDLEY, G. R. HEXT, AND F. R. HIMSWORTH, *Sequential application of simplex design in optimisation and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.

[62] S. SRINIVAS, R. P. FIELD, AND H. .J. HERZOG, *Modeling tar handling options in biomass gasification*, Energy & Fuels, 27 (2013), pp. 2859–2873.

[63] K. SVANBERG, *A class of globally convergent optimization methods based on conservative convex separable approximations*, SIAM: SIAM Journal on Optimization, 12 (2002), pp. 555–573.

[64] V. TORCZON, *On the convergence of the multidirectional search algorithms*, SIAM Journal on Optimization, 1 (1991), pp. 123–145.

[65] ———, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, 7 (1997), pp. 1–25.

[66] S. M. WILD, R. G. REGIS, AND C. A. SHOEMAKER, *ORBIT: optimization by radial basis function interpolation in trust-regions*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3197–3219.

[67] S. M. WILD AND C. SHOEMAKER, *Global convergence of radial basis function trust region derivative-free algorithms*, SIAM Journal on Optimization, 21 (2011), pp. 761–781.