NOWPAC: A PROVABLY CONVERGENT NONLINEAR OPTIMIZER WITH PATH-AUGMENTED CONSTRAINTS FOR NOISY REGIMES

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 \( p \)-reduced fully linear models for the objective function and the constraints. A new constraint-handling scheme based on a quadratic inner boundary path makes the search for feasible trial steps more efficient. In all iterations, the intermediate designs computed by NOWPAC are strictly feasible, and we prove that they 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. For 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, propose an algorithm based on a new constraint-handling scheme, and prove its convergence to a local optimum. 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:

\[
\min f(x) \\
\text{s.t. } c_i(x) \leq 0, \ i = 1, \ldots, r
\]  

(1.1)

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is the objective function defining the quantity of interest and \( c_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \ldots, r \), model the constraints imposed on the design parameters \( x \). The constraints define the set of feasible points \( X := \{ x \in \mathbb{R}^n : c_i(x) \leq 0, \ i = 1, \ldots, r \} \), consisting of all admissible designs. There exist many approaches for approximating the solutions of (1.1); see for instance [4, 5, 8, 12, 29]. 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 [5, 8, 12]. 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, [28]. If the constraints are expensive to evaluate, reduced-order models can be used to reduce the computational costs [2, 18, 26]. 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 [5, 25]. 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 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
2. THE TRUST REGION FRAMEWORK

Typically assume increasing accuracy of the computations while approaching a critical point; see, e.g., [9, 10, 20]. In situations where the objective function and the constraints are only available as black-box evaluations, however, we may not be able to scale the accuracy of these evaluations—i.e., the magnitude of 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 [21] and Nelder and Mead [27] were among the first to propose local direct search methods which only require black-box evaluations of the objective function. This approach was refined in the implicit filtering method by Kelley [23], using simplicial derivatives. [3] and [17] discuss the convergence properties of these sampling-based approaches. Besides sampling-based methods, other derivative-free techniques approximate the objective function using surrogate models. For instance, COBYLA [30, 31] is a widely used algorithm based on a linear surrogate model of the objective. It handles constraints using a penalty approach based on a linear approximation. In [26], a general low-fidelity surrogate model is used for the constraints. However, a rigorous convergence proof for these methods is not available. We refer to [36] for a further overview of derivative-free approaches.

The main contributions of this paper are threefold. First, we present the algorithm NOWPAC (Nonlinear Optimization with Path-Augmented Constraints), which is based on a trust region framework. The algorithm introduces a new way of handling constraints using an “inner boundary path,” which is a quadratic offset to the constraints. The inner boundary path guides the next trial step to become strictly feasible. This strategy alleviates inefficiency caused by the rejection of trial steps that are computed from approximated constraints. Second, we develop a rigorous proof of convergence of the intermediate points computed by NOWPAC to a local 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.

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 [11, 14, 15, 33, 34]. Section 3 describes the algorithm NOWPAC. In Section 4, we prove the convergence of the intermediate points computed by NOWPAC to a first order locally optimal point. The proof is presented under the assumption of accurate evaluations of the objective function and the constraints. In practice, however, we are often faced with irreducible errors in the evaluations. Section 5 thus discusses the behavior of NOWPAC in the latter setting, deriving the asymptotic bounds and error indicator described above. In Section 6, we numerically demonstrate NOWPAC using two test examples and the industrial application of a tar removal process used to produce fuel from biomass. Concluding remarks and a sketch of future work are given in Section 7.

2. The trust region framework. In this section we introduce the derivative-free trust region framework to approximate the solution of the constrained optimization program (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 local critical point $x^*$. For the computation of $x_{k+1}$, trust region methods build surrogates of the objective function $f$ and the constraints $\{c_i\}_{i=1}^m$, denoted by $m_k^f$ and $\{m_k^{c_i}\}_{i=1}^m$ respectively, within a neighborhood of the current point $x_k$. $x_{k+1}$ is determined solely based on the surrogates as being a suitable point that reduces the objective function while staying within the neighborhood of $x_k$ and satisfying the constraints. The neighborhood is called the trust region, $B(x_k, \rho_k) := \{x \in \mathbb{R}^n : \|x - x_k\| \leq \rho_k\}$, with trust region radius $\rho_k$, $k \in \mathbb{N}_0$. Note that there are many possible choices for the surrogates; among them, polynomial response surfaces [14, 32, 33, 35] are widely used. But other approximation methods can be employed as well; for example, radial basis functions are used to create the surrogates in [12]. The particular choice of surrogate models $m_k^f$ and $\{m_k^{c_i}\}_{i=1}^m$ for the objective
function and the constraints is not important in the present context, and we will not go into details on how to compute them. In general, any surrogates that are twice continuously differentiable and satisfy

\[
\begin{align*}
    f(x_k + s) - m^f_i(x_k + s) \leq \kappa_f \rho_k^{2-p} \\
    c_i(x_k + s) - m^c_i(x_k + s) \leq \kappa_c \rho_k^{2-p} \\
    \|\nabla f(x_k + s) - \nabla m^f_i(x_k + s)\| \leq \kappa_{df} \rho_k^{1-p}
\end{align*}
\]  

(2.1a) (2.1b) (2.1c)

with constants \(\kappa_f, \kappa_c, \kappa_{df} > 0\), for all \(x_k + s \in B(x_k, \rho_k), i = 1 \ldots r, p \in [0,1]\) are admissible. In our implementation of NOWPAC, we use a quadratic minimum-Frobenius-norm surrogate; see [35]. Models satisfying (2.1) are called \(p\)-reduced fully linear (\(p\)-RFL) within the trust region \(B(x_k, \rho_k)\). We remark that all of the surrogates previously mentioned satisfy these conditions for all \(p \in [0,1]\), if certain geometry conditions on the sampling points of the model are satisfied; see [13] [35] [72]. Note that the \(p\)-RFL properties stated in (2.1) are a generalization of the 0-RFL properties, which are well known in the literature. The reason for this generalization will become obvious in Section 5. If the reader is not interested in situations with inexact function evaluations, the standard case \(p = 0\) may be assumed in Sections 3 and 4. Without going into details on how to compute the surrogates, we assume that they are denoted by \(m^f_i(x_k + s)\) and \(m^c_i(x_k + s)\), with the corresponding gradients \(g^f_i, g^c_i\) and Hessians \(H^f_i, H^c_i\), \(H^c_i\) at \(s = 0\) for \(i = 1 \ldots r\).

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 2.1.** The objective function \(f\) and the constraints \(\{c_i\}_{i=1}^r\) satisfy:

(a) \(X \cap \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\} \subset X \cap B(x_0, \rho) =: \mathcal{L}\), for \(\rho < \infty\) sufficiently large,

(b) \(f\) and \(c_i\) are continuously differentiable on \(\mathcal{L}\) with Lipschitz continuous gradients,

(c) \(\|\nabla c_i(x)\| \geq \kappa_{bdc}\) for all \(x\) on the boundary of the set \(\mathcal{L}\) and for all \(i = 1 \ldots r\),

(d) the tangential cones on \(\mathcal{L}\) have opening angles greater than \(\sigma > 0\).

**Assumption 2.2.** There exists a constant \(\kappa_{bh} > 0\) such that \(\|H^f_i\| \leq \kappa_{bh} \rho_k^{-p}\), where \(p\) is the order reduction in (2.1).

We note that these assumptions are standard, cf. [11], and ensure the existence of an optimal solution as well as the well-posedness of the optimization problem (1.1). We further remark that, due to the continuity of \(\nabla f\) and \(\{\nabla c_i\}_{i=1}^r\) in Assumption 2.1 we additionally have \(\|\nabla f\| \leq \kappa_{df}\) and \(\|\nabla c_i\| \leq \kappa_{bdc}, i = 1 \ldots r\), for all \(x\) in the compact set \(\mathcal{L}\).

3. The algorithm NOWPAC. In this section we introduce the derivative-free algorithm NOWPAC for the approximation of the solution of the nonlinear constrained problem (1.1). The notation and basic structure follow closely along the lines of [11] [13]; however we introduce significant changes in order to be able to treat the nonlinear constraints as black-box evaluations. A discussion of the algorithm’s convergence is provided in Section 4.

3.1. Preliminaries. We define the extended constraints \(\hat{c}_i(x) := c_i(x) + h_k(x), i = 1 \ldots r\), where the quadratic offset

\[
h_k(x) := \begin{cases} \mathbb{R}^n & \rightarrow \mathbb{R} \\ x & \mapsto \varepsilon_b \|x - x_k\|^2 \end{cases}
\]

is called the inner boundary path. We will justify this perturbation of the constraints within the forthcoming sections. Note that we recover the original constraints by setting the inner boundary

\[\varepsilon_b \rightarrow 0\] for this we require \(\rho_k \leq 1\), which can be achieved by proper rescaling of the design space. However, this always holds in the asymptotic regime \(\rho_k \rightarrow 0\) that we are interested in.
constant $\varepsilon_b$ to zero. 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 locally optimal point:

$$\alpha_k(\rho_k) := \frac{1}{\rho_k} \min_{x_k + d \in X_k \cap B(x_k, \rho_k)} \left\| g_k^f \right\|.$$  

(3.1)

Here, $X_k := \{x \in \mathbb{R}^n : m_k^c(x) + h_k(x) \leq 0\}$ denotes the set of feasible points with respect to the current model of the constraints. Note that the abbreviated notation $c$ represents the corresponding expressions for all $\{c_i\}_{i=1}^r$, e.g. $m_k^c(x) + h_k(x) \leq 0$ means $m_k^c_i(x) + h_k(x) \leq 0$ for all $i = 1 \ldots r$. Besides the criticality measure, we need to specify an initial point $x_0$ and a maximal trust region radius $\rho_{\text{max}} > 0$. Moreover, we set the initial trust region radius to $\rho_0 \in [\rho_{\text{min}}, \rho_{\text{max}}]$, $\rho_{\text{min}} > 0$, and specify technical parameters as shown in Table 3.1. Starting from the current iterate $x_k$, the trust region algorithm computes a trial step $s_k$, such that $x_k + s_k$ minimizes the model $m_k^f$ within $B(x_k, \rho_k) \cap X_k$. We refer to [11] for a discussion on how to efficiently compute a suitable step. We only impose Assumptions [3.1] to hold for the steps computed by Algorithm [10] below, cf. [11].

| symbol | range | description |
|--------|-------|-------------|
| $\varepsilon_b$ | $(0, \infty]$ | inner boundary path constant |
| $\varepsilon_s$ | $(0, \infty]$ | strict feasibility constant |
| $\varepsilon_r$ | $[0, 1)$ | order reduction for strict feasibility |
| $\eta_0$ | $[0, 1]$ | step rejection parameter |
| $\eta_1$ | $[\eta_0, 1]$ | step acceptance parameter\((^{(1)}\)) |
| $\gamma_{\text{inc}}$ | $[1, \infty]$ | increment factor for trust region |
| $\gamma$ | $[0, 1]$ | decrease factor for trust region |
| $\varepsilon_c$ | $[0, \infty]$ | lower bound on trusted criticality measure |
| $\beta$ | $[0, \mu]$ | safety factor |
| $\mu$ | $[\beta, \infty]$ | factor for bound on trust region radius by $\alpha$ |
| $\mu_1, \mu_2, \mu_3$ | $[0, 1]$ | step size parameters\((^{(2)}\)) |

Table 3.1: Technical parameters and their range as used in Algorithm [10]. Additional restrictions: \((^{(1)}\): $\eta_1 > 0$,\n
\((^{(2)}\): $\mu_2 > \mu_1$.\n
**Assumption 3.1.** The trial step $s_k$ computed by Algorithm [7] satisfies

(a) $\left\langle g_k^f, s_k \right\rangle \leq -\mu_3 \alpha_k(\rho_k) \rho_k$,

(b) $m_k^f(x_k + s_k) - m_k^f(x_k) \leq \mu_1 \left\langle g_k^f, s_k \right\rangle$ and

(c) $m_k^f(x_k + s_k) - m_k^f(x_k) \geq \mu_2 \left\langle g_k^f, s_k \right\rangle$,

as well as the feasibility and trust region condition $x_k + s_k \in B(x_k, \rho_k) \cap X_k$.

NOWPAC is designed to work in settings with costly objective function evaluations, dominating over the cost of computing a good trial step $s_k$. This suggests investing more effort in computing a good trial step instead of greedily looking for a crude approximation. Although we prove in the subsequent sections that in the latter case we are still guaranteed to converge to a first order critical point, we expect faster convergence using better trial steps. In our implementation we use the method of moving asymptotes (MMA) [11] as implemented in the NLopt library [22].

**3.2. The algorithm.** We state the complete derivative-free trust region algorithm NOWPAC in Algorithm [10]. For an in-depth discussion of the general framework we refer to [15]. However, we remark that the strict feasibility requirement in STEP 3 of Algorithm [10] cannot be found in the general framework of trust region methods. We address the theoretical importance of the order reduction $\varepsilon_r$,
in the strict feasibility condition $c(x_k + s_k) \leq -\varepsilon_c \rho_k^{2-p-\varepsilon_r}$ in the proof of Lemma 3.3. However, our practical experiences show that NOWPAC is not sensitive to the value of the parameter $\varepsilon_r$.

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

1. Construct the initial $p$-RFL models $m^f_0(x_0 + s)$, $m^c_0(x_0 + s)$.
2. for $k = 0, 1, \ldots$ do
   
   /* STEP 1: Criticality step */
   if $\alpha_k(\rho_k) \leq \varepsilon_c$ then
     if $m^f_k$ and $m^c_k$ are not $p$-RFL in $B(x_k, \rho_k)$ or $\rho_k > \mu \alpha_k(\rho_k)$ then
       Construct $p$-RFL models $\tilde{m}^f_k$ and $\tilde{m}^c_k$ in $B(x_k, \tilde{\rho}_k)$ for some $\tilde{\rho}_k \in [0, \mu \alpha_k(\rho_k)]$
       Set $m^f_k = \tilde{m}^f_k$, $m^c_k = \tilde{m}^c_k$ and $\rho_k = \min\{\max\{\tilde{\rho}_k, \beta \alpha_k(\rho_k)\}, \rho_k\}$
     end
   end

   /* STEP 2: Step calculation */
   Compute a trial step $s_k = \arg\min_{x_k + s \in X_k, \|s\| \leq \rho_k} m^f_k(x_k + s)$ such that Assumptions 3.1 are satisfied

   /* STEP 3: Check strict feasibility of trial point */
   if $c(x_k + s_k) > -\varepsilon_c \rho_k^{2-p-\varepsilon_r}$ then
     Set $\rho_k = \gamma \rho_k$ and update $m^f_k$ and $m^c_k$ accordingly to obtain $p$-RFL models
     Go to line 9
   end

   /* STEP 4: Check acceptance of trial point */
   Compute $r_k = \frac{f(x_k) - f(x_k + s_k)}{m^c_k(x_k) - m^f_k(x_k + s_k)}$
   if $r_k \geq \eta_1$ or both $r_k \geq \eta_0$ and $m^f_k$ is $p$-RFL in $B(x_k, \rho_k)$ then
     Set $x_{k+1} = x_k + s_k$
     Include $x_{k+1}$ into the sample set and update the models to $m^f_{k+1}$ and $m^c_{k+1}$
   else
     Set $x_{k+1} = x_k$, $m^f_{k+1} = m^f_k$ and $m^c_{k+1} = m^c_k$
   end

   /* STEP 5: Model improvement */
   if $r_k < \eta_1$ and $m^f_{k+1}$, $m^c_{k+1}$ are not $p$-RFL then
     Improve the quality of the models $m^f_{k+1}$ and $m^c_{k+1}$
   end

   /* STEP 6: trust region update */
   Set $\rho_{k+1} \in \begin{cases} 
   \{\rho_k, \min\{\gamma \rho_k, \rho_{\text{max}}\}\} & \text{if } r_k \geq \eta_1 \\
   \{\rho_k\} & \text{if } \eta_0 \leq r_k < \eta_1, \\
   \{\gamma \rho_k\} & \text{if } r_k < \eta_0.
   \end{cases}$

end

We call iteration $k$ ‘successful’ if the acceptance ratio $r_k$ exceeds the threshold $\eta_1$, whereas we call it ‘acceptable’ if $\eta_0 \leq r_k < \eta_1$ and the model $m^f_k$ is $p$-RFL. 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 $\rho_{\text{min}} > 0$. We will see in Section 4.2 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

24a | if $\rho_{k+1} < \rho_{\text{min}}$ then stop.
in the actual implementation of NOWPAC. However, we want to examine the asymptotic behavior of the iterates as \( k \to \infty \) and we therefore do not include a stopping criterion in the forthcoming theoretical investigations.

Finally we remark that the construction of \( p \)-RFL models in lines 5, 17, and 22 respectively can be done in a finite number of steps. In particular, the model improvement \textbf{STEP 5} computes at most finitely many intermediate points before the models become \( p \)-RFL; see [15].

3.3. The criticality measure. In case 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

\[
- \nabla f(x^*) \in N(x^*),
\]

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 \( x \). Note that \( N(x) = \{ 0 \} \) whenever \( x \) is an inner point of \( X \) and (3.2) reduces to the gradient criterion \( \| \nabla f(x^*) \| = 0 \) for optimality.

Based on (3.2) we define the exact criticality measure

\[
\alpha[x, \rho] := \frac{1}{\rho} \min_{D(x, \rho)} \langle \nabla f(x), d \rangle,
\]

which gives the maximal possible scaled decrease of the linearized objective function within \( D(x, \rho) := \{ d \in \mathbb{R}^n : x + d \in X, \| d \| \leq \rho \} \).

\textbf{Lemma 3.2.} Under Assumption 2.1, the mapping

\[
\hat{\alpha} : \begin{cases} \mathbb{R}^n \times [0, \infty] \to \mathbb{R} \\ (x, \rho) \mapsto \begin{cases} \frac{1}{\rho} |F_*(x, \rho)| & \text{if } \rho > 0 \\ \lim_{\rho \to 0} \frac{1}{\rho} |F_*(x, \hat{\rho})| & \text{if } \rho = 0 \end{cases} \end{cases}
\]

is continuous, where \( F_* \) is defined by

\[
F_* : \begin{cases} \mathbb{R}^n \times \mathbb{R}_0^+ \to \mathbb{R} \\ (x, \rho) \mapsto \min_{d \in D(x, \rho)} \langle \nabla f(x), d \rangle \end{cases}
\]

\textbf{Proof.} In the case of \( \rho > 0 \), the assertion follows from the continuity of \( \nabla f \). The reasoning uses the continuity results stated in [10] and makes use of the fact that the ratio of two continuous functions is again a continuous function, unless the denominator becomes zero. To complete the proof we have to show that there exists a continuous extension of \( \alpha \) to \( \rho = 0 \). We have

\[
0 \leq |F_*(x, \rho)| = |\langle \nabla f(x), d_*(x, \rho) \rangle| \leq \| \nabla f(x) \| \| d_*(x, \rho) \| \leq \| \nabla f(x) \| \rho,
\]

where \( d_*(x, \rho) := \arg \min_{d \in D(x, \rho)} \langle \nabla f(x), d \rangle \). This shows \( \lim_{\rho \to 0} |F_*(x, \rho)| = 0 \) and

\[
0 \leq \frac{1}{\rho} |F_*(x, \rho)| \leq \| \nabla f(x) \| \leq \kappa_{bdf} \quad \text{for all } \rho \geq 0,
\]

where the case \( \rho = 0 \) is understood in the limit case. Thus, \( \lim_{\rho \to 0} \frac{1}{\rho} |F_*(x, \rho)| \) exists and is finite, and it therefore defines a continuous extension \( \hat{\alpha} \) of \( \alpha \) to the domain \( \mathbb{R}^n \times [0, \infty] \). \( \square \)
In the forthcoming we abuse notation and denote the continuous extension \( \hat{\alpha} \) from Lemma 3.2 with \( \alpha \). We show in Lemma 3.3 that \( \alpha[x, \rho] \) is indeed a measure for criticality.

**Lemma 3.3.** Under Assumption 2.4 the point \( x_\star \in X \) is critical if and only if

\[
\alpha[x_\star, \rho] = 0 \quad \text{for all} \quad \rho \geq 0.
\]

We omit the proof, which follows directly from [11, Lem. 3.3] and the continuity of \( \alpha[x^\star, \rho] \) for \( \rho \to 0 \). At this point, the measure of criticality relies on the gradient \( \nabla f \) of the objective function and the feasible set \( X \). In the context of derivative-free optimization we know neither the gradient nor the exact algebraic structure of the feasible domain. Thus, we are not able to evaluate the criticality measure \( \alpha \) as defined in (3.3). To get a computable criticality measure, we modify \( \alpha \) by replacing \( x \) with \( x_k \) and by substituting the gradient \( g_k^f \) for \( \nabla f(x_k) \). The result is the approximated criticality measure (3.1) used in Algorithm 1. Up to this point, it is not obvious that the approximated criticality measure fits our need to drive the iterates of the algorithm to a critical point of (1.1). However, we show in Lemma 3.4 that the approximated criticality measure \( \alpha_k(\rho_k) \) deviates at most by order \( 1-p \) from the exact criticality measure (3.3), indicating that (3.1) is a reasonable substitute for the exact criticality measure.

**Lemma 3.4.** Under Assumption 2.4 let \( x_k \in X \) be an intermediate point generated by Algorithm 1 and let \( m_k^f \) and \( m_k^c \) be p-RFL models of the objective function \( f \) and constraints \( c \in B(x_k, \rho_k) \) with \( p \in [0, 1] \). Then

\[
|\alpha[x_k, \rho_k] - \alpha_k(\rho_k)| \leq \tilde{c} \rho_k^{1-p}
\]

for \( \rho_k \) sufficiently small and a constant \( \tilde{c} := \bar{c}(\sigma) > 0 \).

*Proof.* We define the intermediate criticality measure, which is based on the gradient \( g_k^f \) of the surrogate model but on the exact feasible domain,

\[
\hat{\alpha}_k(\rho_k) := \frac{1}{\rho_k} \min_{d \in D(x_k, \rho_k)} \langle g_k^f, d \rangle
\]

and use the triangle inequality to get

\[
|\alpha[x_k, \rho_k] - \alpha_k(\rho_k)| \leq |\alpha[x_k, \rho_k] - \hat{\alpha}_k(\rho_k)| + |\hat{\alpha}_k(\rho_k) - \alpha_k(\rho_k)|.
\]

Analogously to [11, Lem. 3.5] we get the estimate \( |\alpha[x_k, \rho_k] - \hat{\alpha}_k(\rho_k)| \leq \kappa_{df} \rho_k^{1-p} \), which constitutes the upper bound on the first term in the right-hand side of (3.4). In order to complete the proof it remains to show that

\[
|\hat{\alpha}_k(\rho_k) - \alpha_k(\rho_k)| \leq \tilde{c} \rho_k^{1-p}
\]

for a constant \( \tilde{c} > 0 \). Using the interpretation of the criticality measure (3.3) as being the maximal possible scaled descent of the linearized objective function within \( D(x, \rho) \), we have to estimate the maximal possible change in the descent of the linearized surrogate of the objective function when replacing the set \( D(x_k, \rho_k) \) by \( \hat{D}(x_k, \rho_k) \).

\[
\hat{D}(x_k, \rho_k) := \{d \in \mathbb{R}^n : m_k^c(x_k + d) + h_k(x_k + d) \leq 0, ||d|| \leq \rho_k\}.
\]

We now look at the maximal possible difference between the sets \( X \) to \( X_k \) in the normal direction to the boundary of \( X \), i.e. we want to derive an upper bound, \( \lambda_x \), of the maximal distance between the level sets \( \{c = 0\} \) and \( \{m_k^c + h_k = 0\} \) in the normal directions \( Vc(\xi) \) for \( \xi \in \{c = 0\} \cap B(x_k, \rho_k) \); cf. left plot in Figure 3.1. Before getting into the details on how to compute \( \lambda_x \), we remark that \( |c(\xi + s) - \nabla c(\xi), s| \leq
3. THE ALGORITHM NOWPAC

\[ \kappa_{tc}\rho_k^2 \] bounding the distance between \( c \) and its tangential space at a boundary point \( \xi \in \{ c = 0 \} \) for any \( s \) such that \( \xi + s \in B(x_k, \rho_k) \). Thus, the triangle inequality yields
\[ |\langle \nabla c(\xi), x - \xi \rangle - m_0^e(x) - h_k(x)| \leq (\kappa_{tc} + \kappa_c\rho_k^p + \varepsilon_b)\rho_k^2. \]

Based on this bound we now derive an upper bound \( \lambda_c \) on the maximal distance of the level sets \( \{ c = 0 \} \) and \( \{ m_0^e + h_k = 0 \} \) in the normal directions \( \nabla c(\xi) \), cf. right plot in Figure 3.1. Since \( \|\nabla c(\xi)\| \geq \kappa_{bdc} \), the angle \( \varphi_1 \) between the tangent (solid line in right plot of Figure 3.1) and the normal, \( \nabla c(\xi) \), (dashed line in right plot of Figure 3.1) at \( \xi \) is not smaller than \( \arctan(\kappa_{bdc}) \), i.e. \( \varphi_1 \geq \arctan(\kappa_{bdc}) \). Now, knowing that \( \lambda_c \leq (\kappa_{tc} + \kappa_c\rho_k^p + \varepsilon_b)\rho_k^2 \), a trigonometric calculation yields
\[ \lambda_c \leq \frac{(\kappa_{tc} + \kappa_c\rho_k^p + \varepsilon_b)\rho_k^2}{\kappa_{bdc}} \leq \kappa_\lambda \rho_k^{-2p} \]
for \( \rho_k \) sufficiently small and \( \kappa_\lambda := (\kappa_{tc} + \kappa_c + \varepsilon_b)\kappa_{bdc}^{-1} \).

Next we relate the distance between \( \hat{\alpha}_k \) and \( \alpha_k \) within the trust region to \( \lambda_c \). For this we note that
\[ \frac{1}{\rho_k} \min_{d \in \hat{D}^{-}(x_k, \rho_k)} \langle g_k^f, d \rangle \leq \hat{\alpha}_k(\rho_k) \leq \frac{1}{\rho_k} \min_{d \in \hat{D}^{+}(x_k, \rho_k)} \langle g_k^f, d \rangle, \]
with
\[ X_k^\pm := \left\{ x \in \mathbb{R}^n : c_i \pm \frac{\nabla c_i}{\|\nabla c_i\|} \lambda_c \leq 0, i = 1 \ldots r \right\} \]
and \( \hat{D}^\pm(x_k, \rho_k) := (X_k^\pm \cap B(x_k, \rho_k)) - \{ x_k \} \). We denote
\[ d_k^+ := \arg \min_{d \in \hat{D}^{+}(x_k, \rho_k)} \langle g_k^f, d \rangle \quad \text{and} \quad d_k^- := \arg \min_{d \in \hat{D}^{-}(x_k, \rho_k)} \langle g_k^f, d \rangle. \]

Thus, we can bound the distance between \( \hat{\alpha}_k \) and \( \alpha_k \) by the distance between \( d_k^+ \) and \( d_k^- \). In case of intersecting constraint functions, see left plot in Figure 3.2 (analogously for the distance between \( \hat{d}_k \) and \( d_k \).
and \(d_k\), we can directly state the bound

\[
|\hat{\alpha}(\rho_k) - \alpha_k(\rho_k)| \leq \frac{1}{\rho_k} \left[ \min_{d \in \mathcal{D}^+(x_k, \rho_k)} \langle g_k^f, d \rangle - \min_{d \in \mathcal{D}^-(x_k, \rho_k)} \langle g_k^f, d \rangle \right] \\
= \frac{1}{\rho_k} \left[ \langle g_k^f, \hat{d}_k \rangle + \langle g_k^f, \hat{d}_k \rangle - \langle g_k^f, d_k^- \rangle \right] \\
\leq \frac{2}{\rho_k} \left\| g_k^f \right\| \kappa_{\lambda} \rho_k^{2-p} \\
\text{for } \rho_k \text{ sufficiently small. However, if } x_k + \hat{d}_k \text{ does not coincide with an intersection point of constraint functions, we can not directly relate the difference between } \hat{\alpha}_k \text{ and } \alpha_k \text{ to } \rho_k^2. \text{ Let } x_k + d_k^+ \text{ be the point on the boundary of } X^- \text{ such that } d_k^- = d_k^+ \text{ is orthogonal to the boundary of } X \text{ and let } x_k + d_k^- \text{ be the point on the boundary of } X \text{ such that } d_k^+ = d_k^- \text{ is orthogonal to the boundary of } X, \text{ cf. right plot in Figure 3.2. Assume for a moment that } x_k + d_k^+ \text{ and } x_k + d_k^- \text{ are contained within the trust region; then}

\[
|\hat{\alpha}(\rho_k) - \alpha_k(\rho_k)| \leq \frac{1}{\rho_k} \left[ \langle g_k^f, d_k^+ \rangle + \langle g_k^f, d_k^- \rangle \right] \\
\leq \frac{2}{\rho_k} \left\| g_k^f \right\| \kappa_{\lambda} \rho_k^{2-p}. \tag{3.5}
\]

Hence, since \(\left\| g_k^f \right\| \leq \left\| \nabla f(x_k) \right\| + \left\| f(x_k) - g_k^f \right\| \leq \kappa_{\text{diff}} + \kappa_{\text{df}} \rho_k^{1-p}\), the assertion of the lemma follows with \(\hat{\sigma}(\sigma) := \kappa_{\text{diff}} + 2 \rho_k^{1-p} \sin(\frac{\pi}{2}) (\kappa_{\text{df}} + \kappa_{\text{df}} \rho_k^{1-p} \max)\) for \(\rho_k\) sufficiently small.

It remains to show that \(x_k + d_k^+\) and \(x_k + \hat{d}_k\) are contained within the trust region; the situation is depicted in Figure 3.3. In the remainder of the proof we will show that this is always the case if \(\rho_k\)
is sufficiently small. To show that $x_k + d_k^\perp$ and $x_k + \hat{d}_k^\perp$ are contained within the trust region we have to ensure that the angles between the vectors $d$ and $\nabla c(x_k + d)$, for $x_k + d \in \{c = 0\} \cap B(x_k, \rho_k)$, are smaller than $\varphi_2 = \arccos \left( \frac{\lambda_x}{2\rho_k} \right)$, i.e.,

$$\frac{\langle \nabla c(x_k + d), d \rangle}{\|\nabla c(x_k + d)\| \|d\|} \geq \frac{\lambda_x}{2\rho_k}.$$  (3.6)

Due to the Lipschitz continuity of $\nabla c$ and $c(x_k + d) = 0$ we have

$$\langle \nabla c(x_k + d), d \rangle = -c(x_k) + \mathcal{O} \left( \rho_k^2 \right) \geq \varepsilon_s \rho_k^{2-p-\varepsilon_r} + \mathcal{O} \left( \rho_k^2 \right) \geq \frac{\varepsilon_s}{\gamma_{inc}} \rho_k^{2-p-\varepsilon_r} + \mathcal{O} \left( \rho_k^2 \right),$$

where $k^*$ is the index of the last acceptable or successful iteration before $k$. Since $\|d\| \leq \rho_k$ and $\|\nabla c(x_k + d)\| \leq \kappa_{ubdc}$, inequality (3.6) holds if

$$\varepsilon_s \rho_k^{2-p-\varepsilon_r} \geq \lambda_x \frac{\kappa_{ubdc}}{2} \gamma_{inc} \rho_k^{2-2p+\varepsilon_r} + \mathcal{O} \left( \rho_k^2 \right),$$

i.e., $\varepsilon_s \geq \kappa_{c} \rho_k^{2-p-\varepsilon_r} + \mathcal{O} \left( \rho_k^{2+\varepsilon_r} \right)$. Thus, $x_k + d_k^\perp$ and $x_k + \hat{d}_k^\perp$ are contained within the trust region if $\rho_k$ is sufficiently small.

In preparation for the convergence proof we need to ensure that the steps $s_k$ computed by NOWPAC provide a sufficiently large descent in the objective function $f$.

**Theorem 3.5.** Consider a sequence $\{x_k\}_k$ generated by NOWPAC and select $k \geq 0$ such that $x_k$ is not critical with respect to the current model, i.e., $\alpha_k(\rho_k) > 0$. Then there exists a constant $\hat{c} \in [0, 1]$ such that

$$m_k^f(x_k) - m_k^f(x_k + s_k) \geq \hat{c} \alpha_k(\rho_k) \min \left\{ \rho_k, \frac{\alpha_k(\rho_k) \rho_k^p}{\kappa_{bh}} \right\}$$

(3.7)

for all $k \geq 0$. 

---

**Fig. 3.3:** Trust region (circle) at $x_k$, the 0 level set of $c$ (solid line) and the envelope of the maximal distances between $X$ and $X_k$ (dotted lines). Left: Two intersections of $\{c = 0\}$ with the boundary of the trust region. The left intersection point violates the angle condition (3.6), while the right intersection point satisfies the angle condition (3.6). Right: Reduced plot only showing $\{c = 0\}$ (solid line) in a neighborhood of the right intersection point $x_k + d$, the vector $d$, the outer normal $\nabla c(x_k + d)$, and the angle $\varphi_2$ between the secant of maximal length $\lambda_x$ as well as the line with direction $d$. 

---
Proof. Applying Taylor’s theorem to the model \( m^f_k \) we get
\[
m^f_k(x_k + s_k) = m^f_k(x_k) + \langle g^f_k, s_k \rangle + \frac{\theta^2}{2} \langle s_k, H^f_k s_k \rangle
\]
for a \( \theta \in [0, 1] \), which under Assumption 2.2 yields
\[
\frac{2}{\|s_k\|^2} \left( m^f_k(x_k + s_k) - m^f_k(x_k) - \langle g^f_k, s_k \rangle \right) \leq \kappa_{bh} \rho_k^{-p}.
\] (3.8)

We conclude from Assumptions 3.1(a) and 3.1(b) that
\[
m^f_k(x_k) - m^f_k(x_k + s_k) \geq -\mu_1 \langle g^f_k, s_k \rangle \geq \mu_1 \alpha_k(\rho_k) \rho_k.
\] (3.9)

Using (3.8) and Assumptions 3.1(a) and 3.1(c) we get
\[
\kappa_{bh} \rho_k^{-p} \geq \frac{2}{\|s_k\|^2} (\mu_2 - 1) \langle g^f_k, s_k \rangle \geq \frac{2}{\rho_k} \mu_3 (1 - \mu_2) \alpha_k(\rho_k),
\] (3.10)
yielding
\[
\rho_k \geq \frac{2 \mu_3 (1 - \mu_2)}{\kappa_{bh} \rho_k^{-p}} \alpha_k(\rho_k).
\] (3.11)

Inserting this into (3.9) results in
\[
m^f_k(x_k) - m^f_k(x_k + s_k) \geq 2 \mu_1 \mu_3^2 (1 - \mu_2) \frac{\alpha_k(\rho_k)^2}{\kappa_{bh} \rho_k^{-p}},
\]
which yields the assertion of the theorem with \( \hat{c} := \mu_1 \mu_3 \min\{1, 2 \mu_3 (1 - \mu_2)\} \).

4. Convergence to first order critical points. In this section we prove convergence of the intermediate points \( \{x_k\}_k \) generated by NOWPAC to a first order critical point of (1.1), i.e.,
\[
\lim_{k \to \infty} \alpha[x_k, \rho_k] = 0.
\]
The general ideas of the proof follow along the lines of [11]. However, although the overall discussions are closely related, we give additional reasoning in order to show convergence in the case of approximated constraint functions. The outline of this section is as follows. In Section 4.1 we show that in every iteration \( k \) there exists a trust region radius \( \rho_k > 0 \) such that NOWPAC accepts the trial step \( s_k \). It turns out that the strict feasibility condition in STEP 3 of NOWPAC plays an important role in guaranteeing progress towards a critical point. Having shown that the algorithm does not get stuck in an infinite loop of rejecting steps, we prove convergence of the intermediate points \( \{x_k\}_k \) to a critical point in Section 4.2.

4.1. Successful iterations. At first, we show that STEP 3 in NOWPAC is successful if the trust region radius is small enough. Thereafter, we take a look at the second hurdle for a successful iteration: the step acceptance condition in STEP 4.

Lemma 4.1. Let \( m_k^c \) be \( p \)-RFL on \( B(x_k, \rho_k) \), \( p \in [0, 1] \), and \( \alpha_k(\rho_k) \geq \kappa_{bh} \rho_k^{1-p} \). In this case STEP 3 in Algorithm 1 is always successful either if

\[
\frac{\kappa_{c}}{\kappa_{bdc}} \rho_k^{e_r} + \varepsilon_s \leq 4 \varepsilon_b \mu_3^2 \rho_k^{p+e_r}
\] (4.1)
or if the trust region radius satisfies

\[ \rho_k \leq \hat{c} \left( \frac{\alpha_k \left( \rho_k \right)}{\kappa_{bh}} \right)^{\frac{1}{1-p}}, \]

where \( k^* \) denotes the last acceptable or successful iteration before \( k \), with the constant

\[ \hat{c} := \left( \frac{\varepsilon_s}{\kappa_{bdc} + \varepsilon_s} \right)^{\frac{1}{1-p}} (2\mu_3(1 - \mu_2))^{\frac{1}{1-p}} < 1. \]

Proof. Note that \textsc{step 3} in Algorithm 1 is successful if the inner boundary path determines a route along the boundary of the feasible domain \( X \) such that \( x_k + s_k \) satisfies the strict feasibility condition, i.e.,

\[ \frac{\kappa_c}{\kappa_{bdc}} \rho_k^{2-p} + \varepsilon_s \rho_k^{2-p-\varepsilon_r} \leq \varepsilon_s \rho_k^{2-p-\varepsilon_r} + \varepsilon_b \| s_k \|^2. \]

Analogously to inequality (3.10) we obtain

\[ \| s_k \| \geq 2\mu_3(1 - \mu_2) \alpha_k \left( \rho_k \right) \geq 2\mu_3(1 - \mu_2) \rho_k, \]

where we used the assumption \( \alpha_k \left( \rho_k \right) \geq \kappa_{bh} \rho_k^{1-p} \). Thus, \textsc{step 3} is successful if the more constraining condition

\[ \frac{\kappa_c}{\kappa_{bdc}} \rho_k^{2-p} + \varepsilon_s \rho_k^{2-p-\varepsilon_r} - 4\varepsilon_b \| s_k \|^2 \leq \varepsilon_s \rho_k^{2-p-\varepsilon_r} \]

is satisfied. Since the right-hand side of this inequality is positive, \textsc{step 3} in Algorithm 1 is successful whenever the left-hand side is smaller than zero, i.e., if (4.1) holds. If (4.1) does not hold we note that all exponents of \( \rho_k \) on the left-hand side of (4.2) are greater than zero, and thus all terms vanish for \( \rho_k \to 0 \). From this we conclude that (4.2) is successful if

\[ \left( \frac{\kappa_c}{\kappa_{bdc}} + \varepsilon_s \right) \rho_k^{2-p-\varepsilon_r} \leq \varepsilon_s \rho_k^{2-p-\varepsilon_r} \]

which, after solving for \( \rho_k \) yields

\[ \rho_k \leq \left( \frac{\kappa_c}{\kappa_{bdc} + \varepsilon_s} \right)^{\frac{1}{1-p}} \rho_k^{1-p}. \]

The assertion of the lemma now follows from (3.11).

From Lemma 4.1 we see that setting the inner boundary path constant \( \varepsilon_b = 0 \) is an admissible choice, without losing the eventual success of \textsc{step 3} in Algorithm 1. However, for practical purposes, Lemma 4.1 suggests choosing \( \varepsilon_b \) sufficiently large in order to have (4.1) satisfied. This results in eliminating any restrictions on the trust region radius imposed by \textsc{step 3} in Algorithm 1. On the other hand, we know from the proof of Lemma 3.4 that a large value of \( \varepsilon_b \) deteriorates the quality of the criticality measure \( \alpha_k \). We also want to point out another drawback of choosing \( \varepsilon_b \) too large. Although Lemma 4.1 guarantees the existence of a successful step \( s_k \), the length \( \| s_k \| \) may be very small. This situation often occurs when the iterates wander along the boundary of \( X_k \). Thus, in practical applications, a choice suited to the objective function at hand should be made. In our practical experience an adaptive choice

\[ \varepsilon_{b,k} := \varepsilon_b \left( \frac{\| s_k - 1 \|}{\rho_k - 1} \right)^2 \]
works very well.

Lemma 4.2. If \( m_k^f \) and \( m_k^e \) are p-RFL on \( B(x_k, \rho_k) \) with order reduction \( p \in [0, 1] \) and \( \rho_k \leq A_k(\alpha_k) \),
\[
A_k(\alpha_k) := \min \left\{ \tilde{c} \frac{\alpha_k(\rho_k)}{\kappa_{bh}}, \frac{\alpha_k(\rho_k)}{\kappa_{bh}} \right\},
\]
where \( k^* \) denotes the last acceptable or successful iteration before \( k \), then the \( k \)-th iteration is successful. The constants \( \tilde{c} \) and \( \tilde{c} \) are the same as in Theorem 4.3 and Lemma 4.1 respectively.

Proof. First, since all assumptions of Lemma 4.2 are satisfied, STEP 3 of Algorithm 1 is successful. Moreover, Theorem 3.5 yields
\[
m^f_k(x_k) - m^f_k(x_k + s_k) \geq \tilde{c} \alpha_k(\rho_k) \rho_k.
\]
Using the p-RFL properties of the model \( m^f_k \) on \( B(x_k, \rho_k) \) we get
\[
|r_k - 1| \leq \frac{|f(x_k + s_k) - m^f_k(x_k + s_k)|}{m^f_k(x_k) - m^f_k(x_k + s_k)} + \frac{|f(x_k) - m^f_k(x_k)|}{m^f_k(x_k) - m^f_k(x_k + s_k)} \leq \frac{2\kappa_f \rho_k^{2-p}}{\tilde{c} \alpha_k(\rho_k) \rho_k} \leq 1 - \eta_1.
\]

4.2. Proof of convergence. Having ensured that NOWPAC always finds a sufficiently small trust region radius to make progress towards a critical point, we now prove the convergence of the intermediate points \( \{x_k\}_k \) to a first order critical point \( x^* \). Following the ideas in [11, 15] we establish a relation between the trust region radii \( \{\rho_k\}_k \) and the criticality measures \( \{\alpha[x_k, \rho_k]\}_k \). We reason that \( \lim_{k \to \infty} \rho_k = 0 \), from which we eventually conclude \( \lim_{k \to \infty} \alpha[x_k, \rho_k] = 0 \). We start by proving the technical auxiliary Lemma 4.3 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 is also bounded from below by a positive constant, cf. [15, Lem. 10.6].

Lemma 4.3. Suppose 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 4.2 (note that STEP 3 in Algorithm 1 ensures p-RFL of the model \( m_k^e \) after every reduction of the trust region radius) it holds that whenever \( \rho_k \) falls below the value
\[
\tilde{\kappa}_2 = A_k(\kappa_1)
\]
the \( k \)-th iteration is either successful or model-improving, and hence it holds that \( \rho_{k+1} \geq \rho_k \). We conclude from this, [11, 15], and the rules of STEPS 1 and 6 that \( \rho_k \geq \min\{\beta \kappa_1, \rho_0, \gamma \tilde{\kappa}_2\} =: \kappa_2 \).

Lemma 4.4. If the number of successful iterations is finite, then
\[
\lim_{k \to \infty} \alpha[x_k, \rho_k] = 0.
\]

Proof. It is proven in [15] that \( \rho_k \to 0 \) for \( k \to \infty \), if the number of successful steps is finite. In the same paper it is shown that \( \|x_j - x_i\| \to 0 \) for \( j \to \infty \), where \( i_j \) denotes the index of the first intermediate point after the \( j \)-th iteration for which the model \( m_j^f \) is p-RFL. Let us now observe that
\[
\alpha[x_j, \rho_j] \leq |\alpha[x_j, \rho_j] - \alpha[x_j, \rho_{i_j}]| + |\alpha[x_j, \rho_{i_j}] - \alpha[x_{i_j}, \rho_{i_j}]| + |\alpha[x_{i_j}, \rho_{i_j}] - \alpha_{i_j}(\rho_{i_j})| + \alpha_{i_j}(\rho_{i_j}).
\]
It remains to show that all four terms on the right-hand side converge to zero. The first two terms converge to zero because of the continuity of $\alpha$ and the facts that $\rho_k \to 0$ and $\|x_j - x_{i_j}\| \to 0$. The third term converges to zero because of Lemma 3.4 and the fact that $m_i^f$ is p-RFL. Finally, the fourth term converges to zero because of Lemma 4.3, since if it were bounded away from zero, then for small enough $\rho_{i_j}$, $i_j$ would be a successful iteration, yielding a contradiction.

So far we have proven our desired convergence result in the case of finitely many successful steps. Now we denote the set of successful steps by $S$ and examine the case of $S$ being a countably infinite set.

**Lemma 4.5.** It holds that

$$\lim_{k \to \infty} \rho_k = 0.$$  

The proof of Lemma 4.5 follows directly from [15, Lem. 10.9] by replacing the stationarity measure therein by $\alpha_k$. Lemma 4.5 shows that using the stopping criterion $\rho_{k+1} < \rho_{\text{min}}$ is reasonable and results in termination of NOWPAC after a finite number of steps. A direct consequence of Lemma 4.5 is

$$\liminf_{k \to \infty} \alpha_k(\rho_k) = 0, \quad (4.4)$$

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 lemma shows that the convergence of a subsequence of the approximated criticality measures $\{\alpha_{k_i}\}$ is carried over to the exact criticality measure.

**Lemma 4.6.** For any subsequence $\{k_i\}$ such that

$$\lim_{i \to \infty} \alpha_{k_i}(\rho_{k_i}) = 0,$$

it also holds that

$$\lim_{i \to \infty} \alpha[x_{k_i}, \rho_{k_i}] = 0.$$  

**Proof.** Due to the convergence of $\alpha_{k_i}(\rho_{k_i}) \to 0$ for $i \to \infty$, there exists an index $j$ such that $\alpha_{k_i}(\rho_{k_i}) \leq \varepsilon_c$ for all $i \geq j$. In this case, STEP 1 of Algorithm ensures that the model $m_i^f$ is p-RFL on $B(x_{k_i}, \rho_{k_i})$ with $\rho_{k_i} \leq \mu \alpha_{k_i}(\rho_{k_i})$ for all $i \geq j$. Using Lemma 3.4 we have

$$|\alpha[x_{k_i}, \rho_{k_i}] - \alpha_{k_i}(\rho_{k_i})| \leq \bar{c} \rho_{k_i}^{1-p} \leq \bar{c} \mu^{1-p} \alpha_{k_i}(\rho_{k_i})^{1-p}.$$  

Thus, the assertion of the lemma follows by using the triangle inequality,

$$\alpha[x_{k_i}, \rho_{k_i}] \leq |\alpha[x_{k_i}, \rho_{k_i}] - \alpha_{k_i}(\rho_{k_i})| + \alpha_{k_i}(\rho_{k_i}).$$

**Theorem 4.7.** It holds that

$$\lim_{k \to \infty} \alpha[x_k, \rho_k] = 0.$$  

**Proof.** Since the theorem holds by Lemma 4.4 if $S$ is finite, we assume that $S$ is infinite in the forthcoming. Following closely along the lines of the proof of [15, Thm. 10.13] we prove the assertion...
by contradiction. Therefore, we assume that there exists a subsequence \( \{ \hat{k}_i \} \) of successful \( (r_{\hat{k}_i} \geq \eta_1) \) iterations such that

\[
\alpha[x_{\hat{k}_i} \rho_{\hat{k}_i}] \geq \varepsilon_0
\]

(4.5)

for some \( \varepsilon_0 > 0 \) for all \( i \). It immediately follows from Lemma 4.3 that

\[
\alpha_{k_i}(\rho_{k_i}) \geq \varepsilon
\]

for some \( \varepsilon > 0 \) for all \( i \) sufficiently large; this especially holds true for

\[
\varepsilon < \min \left\{ \left( \frac{\varepsilon_0}{6 + 2\bar{c}_{\mu}1-p} \right)^{\frac{1}{1-p}}, 1 \right\}.
\]

(4.6)

Based on the subsequence \( \{ \hat{k}_i \} \), we define two subsequences \( \{ k_i \} \) and \( \{ l_i \} \) 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 \( j := \min\{i \in \mathbb{N} : \hat{k}_i > l_i \} \), set \( k_{i+1} = \hat{k}_j \), and choose \( l_{i+1} > k_{i+1} \) as being the first index for which \( \alpha_{l_{i+1}}(\rho_{l_{i+1}}) < \varepsilon \). We remark that the existence of \( \{ l_i \} \) follows from equation (4.4). We thus arrive at the subsequences of indices satisfying

\[
\alpha_{k_i}(\rho_{k_i}) \geq \varepsilon \quad \text{for} \quad k_i \leq k < l_i \quad \text{and} \quad \alpha_{l_i}(\rho_{l_i}) < \varepsilon.
\]

(4.7)

We are now able to conclude the proof of convergence of \( \{ x_{k_i} \} \) to a first order stationary point. The triangle inequality yields

\[
\alpha[x_{k_i}, \rho_{k_i}] \leq |\alpha(x_{k_i}, \rho_{k_i}) - \alpha(x_{l_i}, \rho_{l_i})| + |\alpha(x_{l_i}, \rho_{l_i}) - \alpha(x_{l_i}, \rho_{l_i})| + |\alpha(x_{l_i}, \rho_{l_i}) - \alpha_{l_i}(\rho_{l_i})| + \alpha_{l_i}(\rho_{l_i})
\]

The first two terms of the right-hand side tend to zero because of the continuity of \( \alpha \). The proof of \( \lim_{i \to \infty} \| x_{k_i} - x_l \| = 0 \) is a straightforward extension of the proof of [15] Thm. 10.13 and we omit the details. The first two terms are therefore bounded by \( \varepsilon^{1-p} \) for \( i \) sufficiently large. For the third term we use the fact that from (4.5) and the mechanism of STEP 1, at iteration \( l_i \) the model \( m_{l_i}^f \) is p-RFL on \( B(x_{l_i}, \mu \alpha_{l_i}(\rho_{l_i})) \). Moreover, using Lemma 3.4 and (4.7), we also deduce that the third term is bounded by \( c(\mu \alpha_{l_i}(\rho_{l_i}))^{1-p} < c(\mu \varepsilon)^{1-p} \) for \( i \) sufficiently large. Moreover, the fourth term is bounded by \( \varepsilon^{1-p} \) by (4.7). Summarizing the bounds and using (4.6) results in

\[
\alpha[x_{k_i}, \rho_{k_i}] < (3 + \bar{c} \mu^{1-p}) \varepsilon^{1-p} < \frac{1}{2} \varepsilon_0
\]

for \( i \) large enough, which contradicts (4.5). \( \square \)

5. 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 \( m_k^c \) satisfy the p-RFL properties (2.4). 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 [9] 20 and the references therein) showing convergence in case of increasing accuracy of the evaluations while approaching the optimal design; however, corresponding results for derivative-free methods are missing. Moreover, 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 in order to satisfy the p-RFL properties (2.4). Figure 6.5 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 6.3. 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 [6, 7, 24, 37, 39] 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 2.1 and 2.2. The second terms are the errors. These error terms are only observed 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 \( \delta_{f}^{k} \) and \( \delta_{c}^{k} \). We point out that \( \delta_{f}^{k} \) and \( \delta_{c}^{k} \) 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 \( \delta_{f,max}^{k} \) and \( \delta_{c,max}^{k} \). 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. In order to simplify notation we omit the index \( i \) in the forthcoming. Summarizing, the perturbed observed functions are given by

\[
\begin{align*}
    f_{n,k}(x) &:= f(x) + \delta_{f}^{k}(x), \\
    c_{n,k}(x) &:= c(x) + \delta_{c}^{k}(x).
\end{align*}
\]

In the following theorem we prove the rate of decay—with respect to the trust region radii—that the errors \( \delta_{f}^{k} \) and \( \delta_{c}^{k} \) must obey in order to guarantee convergence of NOWPAC.

**Theorem 5.1.** Consider the \( p \)-RFL minimum Frobenius norm surrogates \( m_{f,n,k}^{k} \) and \( m_{c,n,k}^{k} \) of the observed noisy objective function \( f_{n,k} \) and constraints \( c_{n,k} \), satisfying Assumptions 2.1 and 2.2. The intermediate points \( \{ x_{k} \} \) computed by Algorithm 1 with strict feasibility property

\[
    c(x_{k}) \leq -\varepsilon_{c} - \delta_{c,max}^{k} - \delta_{c,\text{max}}^{k}
\]

converge to a first order critical point if

\[
    \delta_{f,\text{max}}^{k}, \delta_{c,\text{max}}^{k} = o\left( \rho_{k}^{-p} \right).
\]
Proof. In [15, Thm. 5.4] it is shown that

\[
\begin{align*}
|f_{n,k}(x_k + s) - m_{k,n}^{f_{n,k}}(x_k + s)| & \leq \kappa_{f_n} \rho_k^{2-p} \\
|c_{n,k}(x_k + s) - m_{k,n}^{c_{n,k}}(x_k + s)| & \leq \kappa_{c_n} \rho_k^{2-p} \\
\|\nabla f_n(x_k + s) - \nabla m_{k,n}^{f_n}(x_k + s)\| & \leq \kappa_{d_{f_n}} \rho_k^{1-p},
\end{align*}
\]

with

\[
\kappa_{f_n} = \left(\kappa_n + \frac{1}{2}\right) (\nu_f + \|H_k^{f_{n,k}}\|) \rho_k^p \\
\kappa_{c_n} = \left(\kappa_n + \frac{1}{2}\right) (\nu_c + \|H_k^{c_{n,k}}\|) \rho_k^p \\
\kappa_{d_{f_n}} = \kappa_n \left(\nu_f + \|H_k^{f_{n,k}}\|\right) \rho_k^p
\]

(5.3)

where the constant \( \kappa_n \) depends on the geometry of the interpolation points of the minimum Frobenius norm approximation, but does not depend on the trust region radius \( \rho_k \). Here, \( \nu_f, \nu_c, H_k^{f_{n,k}} \) and \( H_k^{c_{n,k}} \) denote the Lipschitz constants of the gradients of \( f_{n,k}, c_{n,k} \) and the Hessians of \( m_{k,n}^{f_{n,k}} \) and \( m_{k,n}^{c_{n,k}} \) respectively. Using the triangle inequality we get

\[
\|\nabla f_{n,k}(x_1) - \nabla f_{n,k}(x_2)\| \leq (\nu_f + \|H_k^{f_{n,k}}\|) \|x_1 - x_2\|,
\]

(\nu_f + \|H_k^{f_{n,k}}\|) \|x_1 - x_2\|,

where \( \nu_f \) and \( \nu_c \) are the Lipschitz constants of \( \nabla f \) and \( \nabla c \), whereas \( H_k^{f_{n,k}} \) and \( H_k^{c_{n,k}} \) denote the Hessians of the error functions \( \delta_{k,f}(x) \) and \( \delta_{k,c}(x) \). Furthermore, it holds that

\[
\|H_k^{f_{n,k}}\| = \|H_k^f + H_k^{f_{n,k}}\| \leq \|H_k^f\| + \|H_k^{f_{n,k}}\| \quad \text{and} \quad \|H_k^{c_{n,k}}\| = \|H_k^c + H_k^{c_{n,k}}\| \leq \|H_k^c\| + \|H_k^{c_{n,k}}\|,
\]

which, together with (5.3), yields

\[
\begin{align*}
\kappa_{f_n} & \leq \left(\kappa_n + \frac{1}{2}\right) (\nu_f + \|H_k^f\| + 2 \|H_k^{f_{n,k}}\|) \rho_k^p \leq \kappa_f^1 \rho_k^p + 2 \kappa_f^0 \delta_{k,max}^f \rho_k^{p-2} \\
\kappa_{c_n} & \leq \left(\kappa_n + \frac{1}{2}\right) (\nu_c + \|H_k^c\| + 2 \|H_k^{c_{n,k}}\|) \rho_k^p \leq \kappa_c^1 \rho_k^p + 2 \kappa_c^0 \delta_{k,max}^c \rho_k^{p-2} \\
\kappa_{d_{f_n}} & \leq \kappa_n \left(\nu_f + \|H_k^f\| + 2 \|H_k^{f_{n,k}}\|\right) \rho_k^p \leq \kappa_d^1 \rho_k^p + \kappa_d^0 \delta_{k,max}^f \rho_k^{p-2}.
\end{align*}
\]

(5.4)

for constants \( \kappa_f^1, \kappa_c^1, \kappa_f^0, \kappa_c^0, \kappa_d^0 > 0 \). In the second inequalities we used

\[
\|H_k^{f_{n,k}}\| \leq \bar{\kappa}_2 \frac{\delta_{k,max}^f}{\rho_k^p} \quad \text{and} \quad \|H_k^{c_{n,k}}\| \leq \bar{\kappa}_2 \frac{\delta_{k,max}^c}{\rho_k^p},
\]

(\text{see [15, Thm. 5.7].}) We remark that the right-hand sides in (5.3) have the same asymptotic behavior as \( \|H_k^f\| \) and \( \|H_k^c\| \) respectively, for \( \|H_k^{f_{n,k}}\| \) and \( \|H_k^{c_{n,k}}\| \) large enough. In order to satisfy the p-RFL
properties (2.1) and Assumption 2.2 we have to ensure that the right-hand sides of (5.1) do not grow unboundedly. Thus, in order for \( \kappa_{f_n}, \kappa_{c_n}, \kappa_{df_n} \) and \( \kappa_{bf}, \rho_k^p \) to be bounded, the errors have to satisfy
\[
\delta^f_{k,\text{max}} \leq \kappa_{\delta^f} \rho_k^{2-p} \quad \text{and} \quad \delta^c_{k,\text{max}} \leq \kappa_{\delta^c} \rho_k^{2-p}
\]
for constants \( \kappa_{\delta^f}, \kappa_{\delta^c} > 0 \).

Note that even though we have access to inexact evaluations of the constraints, we still have to be able to check the strict feasibility property in STEP 3 in Algorithm 1. We account for this by adding the term \( \delta^c_{k,\text{max}} \) in (5.1). However, we point out that in the asymptotic regime, \( k \to \infty, \delta^c_{k,\text{max}} = o(\rho_k^{2-p}) \) will always be dominated by \( \varepsilon_s \rho_k^{2-p-\varepsilon_r} \). Thus, in practical applications, the strict feasibility constant \( \varepsilon_s > 0 \) has to be adapted to the magnitude of the errors (or simply chosen sufficiently large) in order to ensure convergence of Algorithm 1.

We now use Theorem 5.1 to define an indicator that estimates the minimum trust region radius at which further progress of Algorithm 1 is expected to not improve the optimization result. In other words, the indicator detects regimes in which (5.2) is violated. The indicator is based on observing the increase of the norms of \( H_k^f \) or \( H_k^c \) 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 5.1, the norms of \( H_k^f \) and \( H_k^c \) show the same asymptotic behavior as \( \delta^f_{k,\text{max}} \rho_k^{-2} \) and \( \delta^c_{k,\text{max}} \rho_k^{-2} \) for decreasing trust region radii. Thus, on a logarithmic scale, the slope \( \tau \) of the growth of \( \|H_k^f\| \) and \( \|H_k^c\| \) with respect to the trust region radii \( \rho_k \) should not exceed \( p \). 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 \( \kappa_n \) does not grow unboundedly. The latter property is ensured by Algorithm 1 within the model improvement steps. The result is the approximated slope \( \tau (\|H_k\|) \), where \( H_k \) denotes either \( H_k^f \) or \( H_k^c \). The slope can be categorized as indicating a convergent or possibly non-convergent regime, according to the corresponding convergence properties of Algorithm 1; we account for this by adding the term \( \delta^c_{k,\text{max}} \) in (5.1) in order to prevent deterioration of the results.

| Regime                      | \( \tau (\|H_k\|) \) |
|-----------------------------|-----------------------|
| Convergent regime           | \( \tau (\|H_k\|) < p \) |
| Non-convergent regime       | \( \tau (\|H_k\|) \geq p \) |

Table 5.1: Classification of the convergence indicator \( \tau \), which denotes the estimated slope of the Hessian norms with respect to the trust region radii. \( H_k \) denotes either of the Hessians \( H_k^f \) or \( H_k^c \).

6. Numerical results. In this section we apply NOWPAC to three optimization problems. In Sections 6.1 and 6.2 we discuss two model problems: the Rosenbrock function (6.1) and the nonlinear constrained anisotropic exponential example (6.2). 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 5. Thereafter, in Section 6.3 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 for strict feasibility, the inner boundary path, and the error detection scheme to the values shown in Table 6.1. Note that we set the reduction factor \( p \) to a relatively high value. The reason is that, in the current implementation of NOWPAC, we rely on a heuristic to ensure well-poisedness of the points used to build the surrogate models, cf. [33]. This heuristic increases the efficiency of NOWPAC, but at the expense of possibly large constants in the \( p \)-RFL conditions (2.1). When these constants become large,
they deteriorate the pre-asymptotic validity of Theorem 6.1 and we account for this by setting $p$ to the large value of 0.9. 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 pattern search algorithm APPS. We use the implementation of COBYLA \cite{ConnWright2000,Conn2009} from the NLopt optimization library for nonlinear optimization \cite{Lagarisetal2003}. The asynchronous parallel pattern search (APPS) method \cite{Hiejimaetal2018,ConnWright2000} 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 APPS to perform best. We used the APPS code as provided by the hybrid optimization parallel search package (HOPSPACK) \cite{Byrdetal2010}.

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

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

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 a neighborhood around the optimal point $x^*$, and thus constitutes a challenging setting for derivative-free trust region methods like NOWPAC; 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 performance of NOWPAC, COBYLA, and APPS is summarized in Table 6.2. We see that all three methods result in reasonable approximations of the exact optimum. Looking at the number of function evaluations, we see that COBYLA and APPS require slightly more evaluations of the objective function than NOWPAC. The higher number of function evaluations used by APPS is not surprising since it is solely based on sampling, rather than exploiting any approximated gradient information. In addition to the optimal point, NOWPAC computes the criticality measure, which is $\alpha = 1.03 \cdot 10^{-4}$ for stopping threshold $SC = \rho_{\text{min}} = 10^{-3}$, $\alpha = 9.39 \cdot 10^{-5}$ for $SC = \rho_{\text{min}} = 10^{-4}$ and $\alpha = 2.26 \cdot 10^{-6}$ for $SC = \rho_{\text{min}} = 10^{-5}$, reflecting the fact that the approximations are close to the exact critical point. We illustrate the results from Table 6.2 in Figure 6.1 (left) by plotting the distances $d_f$ between the approximated and the exact optimal value against the number of required objective function evaluations. We observe faster convergence of NOWPAC compared to COBYLA and APPS. This is likely due to the fact that NOWPAC uses a minimum Frobenius norm quadratic model, which incorporates second order information about the objective, as opposed to the linear approximations used in COBYLA. Moreover, NOWPAC uses an optimal trial step in local trust region subproblems, which helps to accelerate convergence. For APPS we only report the trend of the decaying distance to the analytical solution by plotting the data given in Table 6.2 as a reference.

Next, we introduce artificial errors into the objective function. The errors are added at every evaluation of the objective function, by randomly drawing values from a uniform distribution on the interval $[-\delta_{\text{max}}^f, \delta_{\text{max}}^f]$ with magnitude $\delta_{\text{max}}^f \in \{10^{-4}, 10^{-3}, 10^{-2}\}$. The corresponding norms of the Hessians $H_k^f$ computed by NOWPAC are shown in Figure 6.2. To create 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 6.3.) We report the distances from the approximated optimal points to the exact optimal design, as well as the corresponding distances for the objective values, in Table 6.3. Here, all distances are computed at the iteration where NOWPAC

| parameter | $\varepsilon_b$ | $\varepsilon_a$ | $\varepsilon_r$ | $p$ |
|-----------|----------------|----------------|---------------|-----|
| value     | 2              | $10^{-3}$      | 0.1           | 0.9 |

Table 6.1: Default parameters of NOWPAC for strict feasibility, the inner boundary path, and the error indicator; used in the test examples of Sections 6.1–6.3.
NUMERICAL RESULTS

|      | \(\mathcal{S}C\) | \#eval | \(x^*\)    | \(f^*\)    | \(d_x\)    | \(d_f\)    |
|------|----------------|--------|------------|------------|------------|------------|
| NOWPAC | \(10^{-3}\) | 60     | (1.01, 1.02)\(^t\) | \(9.47 \cdot 10^{-5}\) | \(2.28 \cdot 10^{-2}\) | \(9.47 \cdot 10^{-5}\) |
| COBYLA | \(10^{-3}\) | 81     | (1.01, 1.02)\(^t\) | \(6.29 \cdot 10^{-5}\) | \(1.81 \cdot 10^{-2}\) | \(6.29 \cdot 10^{-5}\) |
| APPS  | \(10^{-3}\) | 129    | (1.00, 1.01)\(^t\) | \(7.24 \cdot 10^{-6}\) | \(6.12 \cdot 10^{-3}\) | \(7.24 \cdot 10^{-6}\) |
| NOWPAC | \(10^{-4}\) | 109    | (1.00, 1.00)\(^t\) | \(7.69 \cdot 10^{-7}\) | \(2.65 \cdot 10^{-3}\) | \(7.69 \cdot 10^{-7}\) |
| COBYLA | \(10^{-4}\) | 150    | (1.00, 1.00)\(^t\) | \(2.81 \cdot 10^{-7}\) | \(8.47 \cdot 10^{-3}\) | \(2.81 \cdot 10^{-7}\) |
| APPS  | \(10^{-4}\) | 184    | (1.00, 1.00)\(^t\) | \(3.47 \cdot 10^{-8}\) | \(4.07 \cdot 10^{-4}\) | \(3.47 \cdot 10^{-8}\) |
| NOWPAC | \(10^{-5}\) | 140    | (1.00, 1.00)\(^t\) | \(1.06 \cdot 10^{-10}\) | \(2.33 \cdot 10^{-9}\) | \(1.06 \cdot 10^{-10}\) |
| COBYLA | \(10^{-5}\) | 199    | (1.00, 1.00)\(^t\) | \(2.37 \cdot 10^{-9}\) | \(1.05 \cdot 10^{-4}\) | \(2.37 \cdot 10^{-9}\) |
| APPS  | \(10^{-5}\) | 228    | (1.00, 1.00)\(^t\) | \(7.28 \cdot 10^{-10}\) | \(6.35 \cdot 10^{-5}\) | \(7.28 \cdot 10^{-10}\) |

Table 6.2: Summarized performance statistics of NOWPAC, COBYLA, and APPS applied to the Rosenbrock minimization problem 6.1. \(\mathcal{S}C\) indicates the stopping criteria, which is the \(\rho_{\min}\) threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA and APPS. \(d_x\) and \(d_f\) denote the Euclidean distances between the approximated and the analytical solution.

Fig. 6.1: Distances of the approximations computed by NOWPAC, COBYLA, and APPS to the exact optimal value for the Rosenbrock example 6.1 (left) and the exponential example 6.2 (right), dependent on the number of function evaluations. For APPS the dots represent known distances, whereas the dotted line is a linear interpolation.

Fig. 6.2: Norms of the Hessians of the models of the objective function in 6.1 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.

detects the first non-convergent iteration. In the same table, we report the 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. As expected, increasing the magnitude of
6.2 Constrained anisotropic exponential

the errors corrupts the optimal 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, $\delta f_{\text{max}}$. 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 terminate NOWPAC without evaluating many steps that will eventually be rejected.

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

$$\min_{x \in X} - \exp(x^T Dx)$$

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

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

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 $f^* = -1.37 \cdot 10^3$. At the optimal point, the first constraint $c_1(x) = \sin(\|x\|^2) - 0.5$ is active, so that the optimal point is critical, $\alpha(x^*) = 0$, 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.

| $\delta f_{\text{max}}$ | #eval | #saved | $ERR_x$ | $ERR_f$ |
|------------------------|-------|-------|--------|--------|
| $10^{-2}$              | 18    | 15    | $3.32 \cdot 10^{-1}$ | $1.10 \cdot 10^{-2}$ |
| $10^{-3}$              | 29    | 15    | $2.42 \cdot 10^{-1}$ | $8.83 \cdot 10^{-3}$ |
| $10^{-4}$              | 45    | 8     | $7.91 \cdot 10^{-2}$ | $9.72 \cdot 10^{-4}$ |
| 0                     | 140   | 0     | $2.33 \cdot 10^{-5}$ | $1.06 \cdot 10^{-10}$ |

Table 6.3: Summarized performance statistics of NOWPAC applied to the noisy Rosenbrock minimization problem. The stopping criteria is set to $\rho_{\text{min}} = 10^{-5}$. $ERR_x$ and $ERR_f$ denote the error in the Euclidean norm with respect to the analytical solution at early termination. #eval refers to the number of function evaluations at early termination, while #saved is the number of additional evaluations performed by the same run without early termination.

The performance of NOWPAC, COBYLA, and APPS is summarized in Table 6.3. On the one hand, we see the same situation as in the first test example: all three methods produce reasonable approximations of the optimal point and the corresponding minimal value of the objective function. On the other hand, NOWPAC requires fewer evaluations of the objective function than COBYLA or APPS. APPS, in particular, 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.

In Figure 6.1 (right) we plot the distances $d_f$ resulting from running NOWPAC, COBYLA, and APPS up to a specific number of objective function evaluations. The distances for the approximations computed by APPS are not included in the plot for the most part since the number of function evaluations is too large. The computed distances in the runs of COBYLA are very erratic. To understand this behavior we plotted COBYLA’s constraint violations at intermediate designs in Figure 6.1 (dotted
6. NUMERICAL RESULTS

| \( SC \) | \#eval | \( x^* \) | \( f^* \) | \( d_x \) | \( d_f \) |
|--------|--------|--------|--------|--------|--------|
| NOWPAC | \( 10^{-3} \) | 75     | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^3\) | 4.80 \cdot 10^{-3} | 3.01 \cdot 10^{-4} |
| COBYLA | \( 10^{-3} \) | 81     | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^3\) | 2.50 \cdot 10^{-3} | 4.54 \cdot 10^{-4} |
| APPS   | \( 10^{-3} \) | 1580   | \((0, 0, 0.01, -0.03, 0.723)^T\) | \(-1.37 \cdot 10^3\) | 3.24 \cdot 10^{-2} | 1.59 \cdot 10^{-2} |
| NOWPAC | \( 10^{-4} \) | 121    | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^4\) | 1.90 \cdot 10^{-4} | 5.37 \cdot 10^{-7} |
| COBYLA | \( 10^{-4} \) | 130    | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^4\) | 8.74 \cdot 10^{-5} | 1.86 \cdot 10^{-6} |
| APPS   | \( 10^{-4} \) | 4287   | \((0, 0, 0.01, -0.03, 0.723)^T\) | \(-1.37 \cdot 10^4\) | 9.50 \cdot 10^{-3} | 1.29 \cdot 10^{-3} |
| NOWPAC | \( 10^{-5} \) | 136    | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^5\) | 8.14 \cdot 10^{-5} | 9.36 \cdot 10^{-8} |
| COBYLA | \( 10^{-5} \) | 160    | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^5\) | 1.83 \cdot 10^{-5} | 1.14 \cdot 10^{-8} |
| APPS   | \( 10^{-5} \) | 7325   | \((0, 0, 0, 0, 0.724)^T\) | \(-1.37 \cdot 10^5\) | 6.49 \cdot 10^{-1} | 6.44 \cdot 10^{-3} |

Table 6.4: Summarized performance statistics of NOWPAC, COBYLA, and APPS applied to the constrained minimization problem (6.2). \( SC \) indicates the stopping criteria, which is the \( \rho_{min} \) threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA and APPS. \( d_x \) and \( d_f \) denote Euclidean distances between the approximated and the analytical solution.

We see that COBYLA does not respect feasibility of the designs, resulting in spurious reductions for infeasible points at evaluations 33 to 40, for example. In contrast, NOWPAC exhibits a continuous convergence behavior, at the same rate of convergence as COBYLA, while simultaneously guaranteeing strictly feasible intermediate points.

Next, we introduce artificial errors of increasing magnitudes, \( \delta_f^{\max} \in \{10^{-4}, 10^{-3}, 10^{-2}\} \), into the objective function evaluations. The corresponding norms of the Hessians \( H_f^k \) are plotted in Figure 6.3. The plots show that the error indicator classifies the non-convergent iterations very accurately.

![Fig. 6.3: Norms of the Hessians of the models of the objective function in (6.2) for \( \delta_f^{\max} = 10^{-4} \) (left), \( \delta_f^{\max} = 10^{-3} \) (middle) and \( \delta_f^{\max} = 10^{-2} \) (right). Circles indicate the non-convergent regime.](image)

We report on optimization performance and accuracy at early termination in Table 6.5. The overall performance of the error indicator is comparable to its performance in the unconstrained Rosenbrock example—i.e., decreasing the accuracy of the evaluations corrupts the optimal point and reduces the number of objective function evaluations significantly. Note that the distances \( d_f \) correspond roughly to the order of the maximal error magnitude \( \delta_f^{\max} \), resulting in rejected steps which can be avoided by terminating early according to the error indicator.

6.3. Tar removal process model. We now discuss optimization of a tar removal process, shown schematically in Figure 6.4, which is part of the production of synthesis gas (syngas) in a biomass to liquid (BTL) plant [10]. The objective is to maximize the flow rate of purified syngas, \( F_s \), 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, \( x_2 = F_{O_2} \). A call to the

\[\text{NOWPAC}\]
Table 6.5: Summarized performance of the application of NOWPAC to the noisy constrained minimization problem (6.2). The stopping criteria is set to $\rho_{\text{min}} = 10^{-5}$. $d_x$ and $d_f$ denote the Euclidean distances of the approximation to the analytical solution at early termination. $#\text{eval}$ refers to the number of function evaluations at early termination, whereas $#\text{saved}$ is the number of additional evaluations performed by the same run without early termination.

| $\delta^l_{\text{max}}$ | $#\text{eval}$ | $#\text{saved}$ | $ERR_x$ | $ERR_f$ |
|-------------------------|----------------|-----------------|---------|---------|
| $10^{-2}$               | 48             | 11              | $7.82 \cdot 10^{-2}$ | $8.74 \cdot 10^{-2}$ |
| $10^{-3}$               | 75             | 5               | $9.27 \cdot 10^{-3}$ | $1.60 \cdot 10^{-3}$ |
| $10^{-4}$               | 76             | 4               | $4.94 \cdot 10^{-3}$ | $3.37 \cdot 10^{-4}$ |
| 0                       | 136            | 0               | $6.49 \cdot 10^{-7}$ | $6.44 \cdot 10^{-8}$ |

Fig. 6.4: 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.

tar removal process simulator yields the outputs

$$(F_s(x_1, x_2), T_{\text{out}}(x_1, x_2)),$$

where $T_{\text{out}}$ is the temperature of the purified syngas at the outlet. We remark that the removal of tar is implicitly achieved by maximizing $F_s$ 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 $0.5\,[m] \leq x_1 \leq 2\,[m]$ 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 $T_{\text{out}}$ exceeds a limit of 1680 Kelvin, safety concerns impose the constraint on the temperature $T_{\text{out}} \leq 1680\,[K]$. 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

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

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 $T_{\text{out}}$ is only given by black-box evaluations of the tar removal process simulation. Overall, the constrained optimization problem can be stated as

$$\max_{c(x_1, x_2) \leq 0} F_s(x_1, x_2).$$ (6.3)
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 the anisotropy of the elliptical trust region and is tailored to the design space \( X \). We summarize the performance of NOWPAC, COBYLA, and APPS in Table 6.6. Compared to COBYLA, NOWPAC requires fewer evaluations of the objective function for all three tolerances. APPS again requires considerably more objective function evaluations. In all runs, COBYLA proposes designs that are not feasible, i.e., \( T_{out}(x^*) - 1680 = 2.0 \cdot 10^{-5}, T_{out}(x^*) - 1680 = 7.3 \cdot 10^{-3}, \) and \( T_{out}(x^*) - 1680 = 9.9 \cdot 10^{-2} \) for \( SC = 10^{-5} \), \( SC = 10^{-4} \), and \( SC = 10^{-3} \) respectively. This violation of the constraints is not desirable since the reactor may fail when the maximal allowed temperature is exceeded. We note that APPS computes strictly feasible designs in this example.

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 6.3 by plotting the output \((F_t, T_{out})\) for various values of the oxygen inflow rates \( F_{O_2} \in [2.198, 2.200] \) 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 6.3, 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 sharply due to the inexact function evaluations. This behavior shows the effectiveness of the error indicator.

### Table 6.6: Summarized performance statistics of COBYLA, APPS, and NOWPAC in finding an optimal design for the tar removal process model. \( SC \) indicates the stopping criteria, which is the \( \rho_{\min} \) threshold for NOWPAC and the absolute distance in the coordinate directions for COBYLA and APPS.

|       | \( SC \)  | \#eval | \( x^* \)         | \( f^* \)         |
|-------|----------|--------|-------------------|-------------------|
| NOWPAC | \( 10^{-4} \) | 23(7)  | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |
| COBYLA | \( 10^{-3} \) | 24     | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7958 \cdot 10^4 \) |
| APPS   | \( 10^{-3} \) | 102    | \((2.0000, 2.2198 - 10^6)^T\) | \( 3.7956 \cdot 10^4 \) |
| NOWPAC | \( 10^{-4} \) | 23(11) | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |
| COBYLA | \( 10^{-4} \) | 26     | \((2.0000, 2.2201 - 10^6)^T\) | \( 3.7961 \cdot 10^4 \) |
| APPS   | \( 10^{-4} \) | 197    | \((2.0000, 2.2198 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |
| NOWPAC | \( 10^{-5} \) | 23(16) | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |
| COBYLA | \( 10^{-5} \) | 32     | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |
| APPS   | \( 10^{-5} \) | 401    | \((2.0000, 2.2199 - 10^6)^T\) | \( 3.7957 \cdot 10^4 \) |

7. **Conclusions.** This paper has presented a derivative-free trust region method for constrained nonlinear optimization. The method generalizes the work of Conn et al. [11, 15] to handle general black-box constraints without the need for derivative information. We provide a rigorous proof of convergence of the method to first order critical points. This result, given in Section 6.2, 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 p-RFL properties (2.1) of the models \( m^i \) and \( \{m^i_k\}_{i=1} \). In Section 5 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 \( \tau \).
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. Moreover, it is guaranteed to converge to first-order critical points without having to accept errors resulting from approximation of the feasible domain. We emphasize that Algorithm 1 is a skeleton procedure, wherein the user can choose the most appropriate methods for the computation of the trial step and the most suitable approximation method for the surrogates $m_k^f$ and $\{m_k^c\}_{i=1}^r$. 

---

**Fig. 6.5:** Affinely-transformed evaluations of the flowrate of syngas $F_S$ (left) and reactor temperature $T_{out}$ (right) with varying oxygen inflow $F_{O_2} \in [2.2198, 2.2200]$ at reactor lengths of $l = 1.98$ (top), $l = 1.99$ (middle), $l = 2.00$ (bottom).

**Fig. 6.6:** 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.
In future work, we will explore application of the NOWPAC framework to problems in stochastic optimization. As we discussed briefly in Section 5, 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., [38, 6]). 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 $\tau$. 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.

Acknowledgements. This work was supported by BP under the BP-MIT Conversion Research Program. The authors wish to thank their colleagues from the MIT Energy Initiative for providing their simulation code for the tar removal process. The authors also thank Dr. Andrew Conn and Dr. Lior Horesh from the IBM Thomas J. Watson Research Center for helpful discussions and comments on this paper. Moreover, the first author wants to thank Prof. P. Rentrop, Technische Universität München, for an interesting discussion on surrogate models for the trust region subproblems.

REFERENCES

[1] 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.
[2] 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.
[3] C. Audet and J. E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM Journal on Optimization, 17 (2006), pp. 188–217.
[4] D. P. Bertsekas, On the Goldstein-Levitin-Polyak gradient projection method, IEEE Transactions on Automatic Control, 21 (1976), pp. 174–184.
[5] Nonlinear programming, Athena Scientific, Bellmont, MA, 2 ed., 1999.
[6] H. G. Beyer and B. Sendhoff, Robust optimization - a comprehensive survey, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 3190–3218.
[7] 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.
[8] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2009.
[9] 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.
[10] T. D. Choi and C. T. Kelley, Superlinear convergence and implicit filtering, SIAM Journal on Optimization, 10 (2000), pp. 1149–1162.
[11] A. R. Conn, N. Gould, A. Sartenaer, and P. 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.
[12] A. R. Conn, N. I. M. Gould, and P. L. Toint, Trust-region methods, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
[13] 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.
[14] 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.
[15] 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.
[16] A. V. Fiacco and Y. Ishizuka, Sensitivity and stability analysis for nonlinear programming, Annals of Operations Research, 27 (1990), pp. 1–206.
[17] 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.
[18] 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.
[19] 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.
[20] M. Heinkenschloss and L. N. Vicente, Analysis of inexact trust-region SQP algorithms, SIAM Journal on Optimization, 12 (2002), pp. 283–302.
[21] R. Hooke and T. A. Jeeves, "Direct search" solution of numerical and statistical problems, Journal of the ACM, 8 (1961), pp. 212–220.
[22] S. G. Johnson, The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.
[23] C. T. Kelley, Iterative methods for optimization, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 1999.
[24] H. J. Kushner and G. G. Yin, Stochastic approximation algorithms and applications, vol. 35 of Applications of mathematics, Springer Verlag New York, 1997.
[25] X. Liang, H. A. Bashir, and S. Li, Applying infeasible interior point method to SQP for constrained nonlinear programming, International Conference on Computer Science and Software Engineering, 1 (2008), pp. 399–402.
[26] A. March and K. Willcox, Constrained multifidelity optimization using model calibration, Structural and Multidisciplinary Optimization, 46 (2012), pp. 93–109.
[27] J. A. Nelder and R. Mead, A simplex method for function minimization, The Computer Journal, 7 (1965), pp. 308–313.
[28] T. D. Plantenga, HOPSPACK 2.0 user manual, Tech. Report SAND2009-6265, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, October 2009.
[29] M. J. D. Powell, Computing methods in applied sciences and engineering, vol. 704 of Lecture Notes in Mathematics, Springer-Verlag, 1977. ch. Variable metric methods for constrained optimization, pp. 62–72.
[30] 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.
[31] Direct search algorithms for optimization calculations, Acta Numerica, 7 (1998), pp. 287–336.
[32] UOBYQA: Unconstrained optimization by quadratic approximation, Mathematical Programming, Series B, 92 (2002), pp. 555–582.
[33] On trust region methods for unconstrained optimization without derivatives, Mathematical Programming, Series B, 97 (2003), pp. 605–623.
[34] Least Frobenius norm updating of quadratic models that satisfy interpolation conditions, Mathematical Programming, Series B, 100 (2004), pp. 183–215.
[35] L. M. Rios and N. V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, Journal of Global Optimization, 56 (2012), pp. 1247–1293.
[36] H. Robbins and S. Monro, A stochastic approximation method, Annals of Mathematical Statistics, 22 (1951), pp. 400–407.
[37] 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.
[38] A. Shapiro, D. Dentcheva, and A. Ruszczyński, Lectures on stochastic programming, Society for Industrial and Applied Mathematics and the Mathematical Programming Society, 2009.
[39] S. Srinivas, R. P. Field, and H. J. Herzog, Modeling tar handling options in biomass gasification, Energy & Fuels, 27 (2013), pp. 2859–2873.
[40] K. Svärdberg, A class of globally convergent optimization methods based on conservative convex separable approximations, SIAM: SIAM Journal on Optimization, 12 (2002), pp. 555–573.
[41] 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.