SIAM J. OPTIM.
Vol. 28, No. 2, pp. 1533–1563

COMPLEXITY ANALYSIS OF A TRUST FUNNEL ALGORITHM FOR EQUALITY CONSTRAINED OPTIMIZATION*

FRANK E. CURTIS†, DANIEL P. ROBINSON‡, AND MOHAMMADREZA SAMADI†

Abstract. A method is proposed for solving equality constrained nonlinear optimization problems involving twice continuously differentiable functions. The method employs a trust funnel approach consisting of two phases: a first phase to locate an ϵ-feasible point and a second phase to seek optimality while maintaining at least ϵ-feasibility. Two-phase approaches of this kind based on a cubic regularization methodology have recently been proposed along with supporting worst-case iteration complexity analyses. Notably, in these approaches, the objective function is completely ignored in the first phase when ϵ-feasibility is sought. The main contribution of the method proposed in this paper is that the same worst-case iteration complexity is achieved, but with a first phase that also accounts for improvements in the objective function. As such, the method attempts to put significantly less burden on the second phase for seeking optimality.

Key words. equality constrained optimization, nonlinear optimization, nonconvex optimization, trust funnel methods, worst-case iteration complexity

AMS subject classifications. 49M15, 49M37, 65K05, 65K10, 65Y20, 68Q25, 90C30, 90C60

DOI. 10.1137/16M1108650

1. Introduction.

The purpose of this paper is to propose a new method for solving equality constrained nonlinear optimization problems. As is well known, such problems are important throughout science and engineering, arising in areas such as network flow optimization [26, 33], optimal allocation with resource constraints [13, 27], maximum likelihood estimations with constraints [25], and optimization with constraints defined by partial differential equations [1, 3, 34].

Contemporary methods for solving equality constrained optimization problems are predominantly based on ideas of sequential quadratic optimization (commonly known as SQP) [4, 15, 16, 20, 21, 22, 29, 32]. The design of such methods remains an active area of research as algorithm developers aim to propose new methods that attain global convergence guarantees under weak assumptions about the problem functions. Recently, however, researchers are being drawn to the idea of designing algorithms that also offer improved worst-case iteration complexity bounds; see, e.g., [7]. This is due to the fact that, at least for convex optimization, algorithms designed with complexity bounds in mind have led to methods with improved practical performance.

For solving equality constrained problems, a cubic regularization method is proposed in [10] with an eye toward achieving good complexity properties. This is a two-phase approach with a first phase that seeks an ϵ-feasible point and a second phase that seeks optimality while maintaining ϵ-feasibility. The number of iterations that the method requires in the first phase is $\mathcal{O}(\epsilon^{-3/2})$, a bound that is known to

*Received by the editors December 20, 2016; accepted for publication (in revised form) February 12, 2018; published electronically May 8, 2018.
http://www.siam.org/journals/siopt/28-2/M110865.html
Funding: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics, Early Career Research Program under contract number DE–SC0010615, as well as by the U.S. National Science Foundation under grant DMS–1319356.
†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (frank.e.curtis@gmail.com, mos213@lehigh.edu).
‡Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218 (daniel.p.robinson@jhu.edu).

be optimal for unconstrained optimization [8]. The authors of [10] then propose a method for the second phase and analyze its complexity properties. Such a two-phase approach is analyzed further—with a careful emphasis on termination conditions for each phase—in [2]. (For related work on cubic regularization methods, see [9, 11].)

Notably, the methods in [2, 10] represent a departure from the current state-of-the-art SQP methods that offer the best practical performance; see also [28]. One of the main reasons for this is that contemporary SQP methods seek feasibility and optimality *simultaneously*. By contrast, the approaches from [2, 10] might not offer practical benefits due to the fact that the first phase of each algorithm entirely ignores the objective function, meaning that numerous iterations might need to be performed before the objective function influences the trajectory of the algorithm.

The algorithm proposed in this paper can be considered a next step in the design of *practical* algorithms for equality constrained optimization with good worst-case iteration complexity properties. Ours is also a two-phase approach, but is closer to the SQP-type methods representing the state-of-the-art for solving equality constrained problems. In particular, the first phase of our proposed approach follows a trust funnel methodology that locates an ϵ-feasible point in $\mathcal{O}(\epsilon^{-3/2})$ iterations *while also attempting to yield improvements in the objective function.* Borrowing ideas from the trust region method known as TRACE [17], we prove that our method attains the same worst-case iteration complexity bounds as those offered by [2, 10].

Organization. In the rest of this section, we introduce notation used throughout the paper and cover preliminary material on equality constrained nonlinear optimization. In section 2, we motivate and describe our proposed "phase 1" method for locating an ϵ-feasible point while also attempting to reduce the objective function. An analysis of the convergence and worst-case iteration complexity of this phase 1 method is presented in section 3. Strategies and convergence/complexity guarantees for "phase 2" are given in section 4. Numerical results are given in section 5. Concluding remarks are given in section 6.

Notation. A vector with all elements equal to 1 is denoted by e and an identity matrix is denoted by I, where, in each case, the size of the quantity is determined by the context in which it appears. With real symmetric matrices A and B, let $A \succ (\succeq)$ B indicate that $A - B$ is positive definite (semidefinite); e.g., $A \succ (\succeq)\ 0$ indicates that A is positive definite (semidefinite). Given vectors $(u, v) \in \mathbb{R}^N \times \mathbb{R}^N$, let $u \perp v$ mean that $u_i v_i = 0$ for all $i \in \{1, 2, \ldots, N\}$. Let $\|x\|$ denote the ℓ_2-norm of x.

1.1. Preliminaries. Given an objective function $f : \mathbb{R}^N \to \mathbb{R}$ and constraint function $c : \mathbb{R}^N \to \mathbb{R}^M$, we study the equality constrained optimization problem

$$(1) \qquad \min_{x \in \mathbb{R}^N} \ f(x) \quad \text{subject to (s.t.)} \quad c(x) = 0,$$

where "s.t." indicates "subject to." At the outset, let us state the following assumption about the problem functions.

Assumption 1. The functions f and c are twice continuously differentiable.

In light of Assumption 1, we define $g : \mathbb{R}^N \to \mathbb{R}^N$ as the gradient function of f, i.e., $g := \nabla f$, and define $J : \mathbb{R}^N \to \mathbb{R}^{M \times N}$ as the Jacobian function of c, i.e., $J := \nabla c^T$. The function $c_i : \mathbb{R}^N \to \mathbb{R}$ denotes the ith element of the function c.

Our proposed algorithm follows a local search strategy that merely aims to compute a first-order stationary point for problem (1). Defining the Lagrangian $\mathcal{L}$: $\mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}$ as given by $\mathcal{L}(x, y) = f(x) + y^T c(x)$, a first-order stationary point (x, y) is one that satisfies $0 = \nabla_x L(x, y) \equiv g(x) + J(x)^T y$ and $0 = \nabla_y L(x, y) \equiv c(x)$.

Our proposed technique for solving problem (1) is iterative, generating, amongst other quantities, a sequence of iterates $\{x_k\}$ indexed by $k \in \mathbb{N}$. For ease of exposition, we also apply an iteration index subscript for the function and other quantities corresponding to the kth iteration; e.g., we write f_k to denote $f(x_k)$.

2. Phase 1: Obtaining approximate feasibility.

The goal of phase 1 is to obtain an iterate that is (approximately) feasible. This can, of course, be accomplished by employing an algorithm that focuses exclusively on minimizing a measure of constraint violation. However, we find this idea to be unsatisfactory since such an approach would entirely ignore the objective function. As an alternative, in this section, we present a trust funnel algorithm with good complexity properties for obtaining (approximate) feasibility that attempts to simultaneously reduce the objective f.

2.1. Step computation.

Similar to other trust funnel algorithms [23, 14], our algorithm employs a step-decomposition approach wherein each trial step is composed of a *normal step* aimed at reducing constraint violation (i.e., infeasibility) and a *tangential step* aimed at reducing the objective function. The algorithm then uses computed information, such as the reductions that the trial step yields in models of the constraint violation and objective function, to determine which of two types of criteria should be used for accepting or rejecting the trial step. To ensure that sufficient priority is given to obtaining (approximate) feasibility, an upper bound on a constraint violation measure is initialized, maintained, and subsequently driven toward zero as improvements toward feasibility are obtained. The algorithm might also nullify the tangential component of a trial step, even after it is computed, if it is deemed too harmful in the algorithm's pursuit toward (approximate) feasibility.

2.1.1. Normal step.

The purpose of the normal step is to reduce infeasibility. The measure of infeasibility that we employ is $v : \mathbb{R}^N \to \mathbb{R}$, defined by

$$v(x) = \tfrac{1}{2}\|c(x)\|^2. \tag{2}$$

At an iterate x_k, the normal step n_k is defined as a minimizer of a second-order Taylor series approximation of v at x_k subject to a trust region constraint, i.e.,

$$n_k \in \arg\min_{n\in\mathbb{R}^N} m_k^v(n) \quad \text{s.t.} \quad \|n\| \le \delta_k^v, \tag{3}$$

where the scalar $\delta_k^v \in (0,\infty)$ is the trust region radius and the model of the constraint violation measure at x_k is $m_k^v : \mathbb{R}^N \to \mathbb{R}$, defined by

$$m_k^v(n) = v_k + g_k^{vT} n + \tfrac{1}{2} n^T H_k^v n \quad \text{with} \quad g_k^v := \nabla v(x_k) = J_k^T c_k, \tag{4}$$

$$\text{and} \quad H_k^v := \nabla^2 v(x_k) = J_k^T J_k + \sum_{i=1}^M c_i(x_k)\nabla^2 c_i(x_k). \tag{5}$$

For any $(x_k, \delta_k^v) \in \mathbb{R}^N \times \mathbb{R}_+$, a globally optimal solution to (3) exists [12, Corollary 7.2.2] and n_k has a corresponding dual variable $\lambda_k^v \in \mathbb{R}_+$ such that

$$g_k^v + (H_k^v + \lambda_k^v I) n_k = 0, \tag{6a}$$

$$H_k^v + \lambda_k^v I \succeq 0, \tag{6b}$$

$$\text{and} \quad 0 \le \lambda_k^v \perp (\delta_k^v - \|n_k\|) \ge 0. \tag{6c}$$

In a standard trust region strategy, a trust region radius is given at the beginning of an iteration, which *explicitly* affects the solution of the subproblem. Indeed, for ease of exposition and analysis, our method is stated in this manner. However, for the purposes of implementation, one might recognize that our method could, in some circumstances—specifically, after any time the normal step trust region radius is contracted—compute a normal step as a solution of (3) where the radius is defined *implicitly* by a given dual variable λ_k^v. In particular, given a $\lambda_k^v \in [0,\infty)$ that is strictly larger than the negative of the leftmost eigenvalue of H_k^v, one could compute n_k from

$$\mathcal{Q}_k^v(\lambda_k^v): \quad \min_{n\in\mathbb{R}^N} \; v_k + g_k^{vT} n + \tfrac{1}{2} n^T (H_k^v + \lambda_k^v I) n. \tag{7}$$

The unique solution to (7), call it $n_k(\lambda_k^v)$, is the solution of the nonsingular linear system $(H_k^v + \lambda_k^v I)n = -g_k^v$, and is the global solution of (3) for $\delta_k^v = \|n_k(\lambda_k^v)\|$.

2.1.2. Tangential step. The purpose of the tangential step is to reduce the objective function. Specifically, when requested by the algorithm, the tangential step t_k is defined as a minimizer of a quadratic model of the objective function in the null space of the constraint Jacobian subject to a trust region constraint, i.e.,

$$t_k \in \arg\min_{t\in\mathbb{R}^N} m_k^f(n_k + t) \quad \text{s.t.} \quad J_k t = 0 \quad \text{and} \quad \|n_k + t\| \le \delta_k^s, \tag{8}$$

where $\delta_k^s \in (0,\infty)$ is a trust region radius and, with some symmetric $H_k \in \mathbb{R}^{N\times N}$, the objective function model $m_k^f : \mathbb{R}^N \to \mathbb{R}$ is defined by

$$m_k^f(s) = f_k + g_k^T s + \tfrac{1}{2} s^T H_k s. \tag{9}$$

Following other trust funnel strategies, one desires δ_k^s to be set such that the trust region describes the area in which the models of the constraint violation and objective function are accurate. In particular, with a trust region radius $\delta_k^f \in (0,\infty)$ for the objective function, our algorithm employs, for some scalar $\kappa_\delta \in (1,\infty)$, the value

$$\delta_k^s := \min\{\kappa_\delta \delta_k^v, \delta_k^f\}. \tag{10}$$

Due to this choice of trust region radius, it is deemed not worthwhile to compute a nonzero tangential step if the feasible region of (8) is small. Specifically, our algorithm only computes a nonzero t_k when $\|n_k\| \le \kappa_n \delta_k^s$ for some $\kappa_n \in (0,1)$. In addition, it only makes sense to compute a tangential step when reasonable progress in reducing f in the null space of J_k can be expected. To predict the potential progress, we define

$$g_k^p := Z_k Z_k^T (g_k + H_k n_k), \tag{11}$$

where the columns of Z_k form an orthonormal basis for $\mathrm{Null}(J_k)$. If $\|g_k^p\| < \kappa_p \|g_k^v\|$ for some $\kappa_p \in (0,\infty)$, then computing a tangential step is not worthwhile and we simply set the primal-dual solution (estimate) for (8) to zero.

For any $(x_k, \delta_k^s, H_k) \in \mathbb{R}^N \times \mathbb{R}_+ \times \mathbb{R}^{N\times N}$, a globally optimal solution to (8) exists [12, Corollary 7.2.2] and t_k has corresponding dual variables $y_k^f \in \mathbb{R}^M$ and $\lambda_k^f \in \mathbb{R}_+$ (for the null space and trust region constraints, respectively) such that

$$\begin{bmatrix} H_k + \lambda_k^f I & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} t_k \\ y_k^f \end{bmatrix} = - \begin{bmatrix} g_k + (H_k + \lambda_k^f I) n_k \\ 0 \end{bmatrix}, \tag{12a}$$

$$Z_k^T H_k Z_k + \lambda_k^f I \succeq 0, \tag{12b}$$

$$\text{and} \quad 0 \le \lambda_k^f \perp (\delta_k^s - \|n_k + t_k\|) \ge 0. \tag{12c}$$

As for the normal step computation, an implementation of our algorithm could compute t_k not as a solution of (8) for a given δ_k^s, but as a solution of a regularized subproblem for a given dual variable for the trust region constraint. Specifically, for a $\lambda_k^f \in [0, \infty)$ that is strictly larger than the negative of the leftmost eigenvalue of $Z_k^T H_k Z_k$, one could solve the following subproblem for the tangential step:

$$\text{(13)} \quad \mathcal{Q}_k^f(\lambda_k^f): \quad \min_{t \in \mathbb{R}^N} \; (g_k + (H_k + \lambda_k^f I) n_k)^T t + \tfrac{1}{2} t^T (H_k + \lambda_k^f I) t \quad \text{s.t.} \quad J_k t = 0.$$

The unique solution $t_k(\lambda_k^f)$ of (13) is a global solution of (8) for $\delta_k^s = \|n_k + t_k(\lambda_k^f)\|$.

There are situations in which our algorithm discards a computed tangential step after one is computed, i.e., situations when the algorithm resets $t_k \leftarrow 0$. Specifically, this occurs when any of the following conditions fails to hold:

$$\text{(14a)} \quad m_k^v(0) - m_k^v(n_k + t_k) \geq \kappa_{vm} \left(m_k^v(0) - m_k^v(n_k) \right) \quad \text{for some} \quad \kappa_{vm} \in (0, 1),$$

$$\text{(14b)} \quad \|n_k + t_k\| \geq \kappa_{ntn} \|n_k\| \quad \text{for some} \quad \kappa_{ntn} \in (0, 1),$$

$$\text{(14c)} \quad \|H_k^v t_k\| \leq \kappa_{ht} \|n_k + t_k\|^2 \quad \text{for some} \quad \kappa_{ht} \in (0, \infty).$$

The first of these conditions requires that the reduction in the constraint violation model for the full step $s_k := n_k + t_k$ is sufficiently large with respect to that obtained by the normal step; the second requires that the full step is sufficiently large in norm compared to the normal step; and the third requires that the action of the tangential step on the Hessian of the constraint violation model is not too large compared to the squared norm of the full step. It is worthwhile to mention that all of these conditions are satisfied automatically when $H_k^v = J_k^T J_k$ (recall (5)), which occurs, e.g., when c is affine. However, since this does not hold in general, our algorithm requires these conditions explicitly, or else resets the tangential step to zero (which satisfies (14)).

2.2. Step acceptance. After computing a normal step n_k and potentially a tangential step t_k, the algorithm determines whether to accept the full step $s_k := n_k + t_k$. The strategy that it employs is based on first using the obtained reductions in the models of constraint violation and the objective, as well as other related quantities, to determine what should be the main goal of the iteration: reducing constraint violation or the objective function. Since the primary goal of phase 1 is to obtain (approximate) feasibility, the algorithm has a preference toward reducing constraint violation unless the potential reduction in the objective function is particularly compelling. Specifically, if the following conditions hold, indicating good potential progress in reducing the objective, then the algorithm performs an F-ITERATION (see section 2.2.1):

$$\text{(15a)} \quad t_k \neq 0 \quad \text{with} \quad \|t_k\| \geq \kappa_{st} \|s_k\| \quad \text{for some } \kappa_{st} \in (0, 1),$$

$$\text{(15b)} \quad m_k^f(0) - m_k^f(s_k) \geq \kappa_{fm} (m_k^f(n_k) - m_k^f(s_k)) \quad \text{for some } \kappa_{fm} \in (0, 1),$$

$$\text{(15c)} \quad v(x_k + s_k) \leq v_k^{\max} - \kappa_\rho \|s_k\|^3 \quad \text{for some } \kappa_\rho \in (0, 1),$$

$$\text{(15d)} \quad n_k^T t_k \geq -\tfrac{1}{2} \kappa_{ntt} \|t_k\|^2 \quad \text{for some } \kappa_{ntt} \in (0, 1),$$

$$\text{(15e)} \quad \lambda_k^v \leq \sigma_k^v \|n_k\| \quad \text{and}$$

$$\text{(15f)} \quad \left\| (H_k - \nabla^2 f(x_k)) s_k \right\| \leq \kappa_{hs} \|s_k\|^2 \quad \text{for some } \kappa_{hs} \in (0, \infty).$$

Conditions (15a)–(15c) are similar to those employed in other trust funnel algorithms, except that (15a) and (15c) are stronger (than the common, weaker requirements that $t_k \neq 0$ and $v(x_k + s_k) \leq v_k^{\max}$). Employed here is a scalar sequence $\{v_k^{\max}\}$ updated dynamically by the algorithm that represents an upper bound on constraint violation; for this sequence, the algorithm ensures (see Lemma 6) that $v_k \le v_k^{\max}$ and $v_{k+1}^{\max} \le v_k^{\max}$ for all $k \in \mathbb{N}$. Condition (15d) ensures that, for an F-ITERATION, the inner product between the normal and tangential steps is not too negative (or else the tangential step might undo too much of the progress toward feasibility offered by the normal step). Finally, conditions (15e) and (15f) are essential for achieving good complexity properties, requiring that any F-ITERATION involves a normal step that is sufficiently large compared to the Lagrange multiplier for the trust region constraint and that the action of the full step on H_k does not differ too much from its action on $\nabla^2 f(x_k)$. If any condition in (15) does not hold, then a V-ITERATION is perfomed (see section 2.2.2).

Viewing (15f), it is worthwhile to reflect on the choice of H_k in the algorithm. With the attainment of optimality (not only feasibility) in mind, standard practice would suggest that it is desirable to choose H_k as the Hessian of the Lagrangian of problem (1) for some multiplier vector $y_k \in \mathbb{R}^M$. This multiplier vector could be obtained, e.g., as the *QP multipliers* from some previous iteration or *least squares multipliers* using current derivative information. For our purposes of obtaining good complexity properties for phase 1, we do not require a particular choice of H_k, but this discussion and (15f) do offer some guidance. Specifically, one might choose H_k as an approximation of the Hessian of the Lagrangian, potentially with the magnitude of the multiplier vector restricted in such a way that, after the full step is computed, the action of it on H_k will not differ too much with its action on $\nabla^2 f(x_k)$. This is more reasonable to do when it is known that the set of optimal multipliers is bounded (which is true, e.g., under various constraint qualifications). In any case, setting H_k to $\nabla f^2(x_k)$ is at least one valid choice as far as our analysis is concerned.

2.2.1. F-iteration. If (15) holds, then we determine that the kth iteration is an F-ITERATION. In this case, we begin by calculating the quantity

$$\rho_k^f \leftarrow (f_k - f(x_k + s_k))/\|s_k\|^3, \tag{16}$$

which measures decrease in f. Using this quantity, acceptance or rejection of the step and the rules for updating the trust region radius are similar to those in [17]. As for the trust funnel radius, rather than the update in [23, Algorithm 2.1], we require a modified update; in particular, we use, for some $\{\kappa_{v1}, \kappa_{v2}\} \subset (0,1)$,

$$v_{k+1}^{\max} \leftarrow \min\{\max\{\kappa_{v1} v_k^{\max}, v_k^{\max} - \kappa_\rho \|s_k\|^3\}, v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1})\}. \tag{17}$$

2.2.2. V-iteration. When any one of the conditions in (15) does not hold, the kth iteration is a V-ITERATION, during which the main focus is to decrease the measure of infeasibility v. In this case, we calculate

$$\rho_k^v \leftarrow (v(x_k) - v(x_k + s_k))/\|s_k\|^3, \tag{18}$$

which provides a measure of the decrease in constraint violation. The rules for accepting or rejecting the trial step and for updating the trust region radius are the same as those in [17]. One addition is that during a successful V-ITERATION, the trust funnel radius is updated, using the same constants as in (17), as

$$v_{k+1}^{\max} \leftarrow \min\{\max\{\kappa_{v1} v_k^{\max}, v_{k+1} + \kappa_{v2}(v_k - v_{k+1})\}, v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1})\}. \tag{19}$$

2.3. Algorithm statement. Our complete algorithm for finding an (approximately) feasible point can now be stated as Algorithm 1 on page 1539, which in turn calls the F-ITERATION subroutine stated as Algorithm 2 on page 1540 and the V-ITERATION subroutine stated as Algorithm 3 on page 1541.

Algorithm 1 Trust funnel algorithm for phase 1

Require: $\{\kappa_n, \kappa_{vm}, \kappa_{ntn}, \kappa_\rho, \kappa_{fm}, \kappa_{st}, \kappa_{ntt}, \kappa_{v1}, \kappa_{v2}, \gamma_c\} \subset (0,1)$,
$\{\kappa_p, \kappa_{ht}, \kappa_{hs}, \epsilon, \underline{\sigma}\} \subset (0,\infty)$, $\{\kappa_\delta, \gamma_e, \gamma_\lambda\} \in (1,\infty)$, and $\overline{\sigma} \in [\underline{\sigma}, \infty)$;
F-ITERATION (Algorithm 2, page 1540) and V-ITERATION (Algorithm 3, page 1541)

procedure TRUST_FUNNEL
  choose $x_0 \in \mathbb{R}^N$, $v_0^{\max} \in [\max\{1, v_0\}, \infty)$, and $\sigma_0^v \in [\underline{\sigma}, \overline{\sigma}]$
  choose $\{\delta_0^v, \Delta_0^v, \delta_0^f\} \subset (0,\infty)$ such that $\delta_0^v \le \Delta_0^v$
  for $k \in \mathbb{N}$ **do**
    if $\|g_k^v\| \le \epsilon$ **then return** x_k
    $(n_k, t_k, \lambda_k^v, \lambda_k^f) \leftarrow$ COMPUTE_STEPS$(x_k, \delta_k^v, \delta_k^s)$
    set $s_k \leftarrow n_k + t_k$
    set $\sigma_k^v \leftarrow$ COMPUTE_SIGMA$(n_k, \lambda_k^v, \sigma_{k-1}^v, \rho_{k-1}^v)$
    if (15) is satisfied **then**
      set ρ_k^f by (16)
      $(x_{k+1}, v_{k+1}^{\max}, \delta_{k+1}^f) \leftarrow$ F-ITERATION$(x_k, n_k, s_k, v_k^{\max}, \delta_k^f, \lambda_k^f, \rho_k^f)$
      set $\delta_{k+1}^v \leftarrow \delta_k^v$, $\Delta_{k+1}^v \leftarrow \Delta_k^v$, and $\rho_k^v \leftarrow \infty$
    else
      set ρ_k^v by (18)
      $(x_{k+1}, v_{k+1}^{\max}, \delta_{k+1}^v, \Delta_{k+1}^v) \leftarrow$ V-ITERATION$(x_k, n_k, s_k, v_k^{\max}, \delta_k^v, \Delta_k^v, \lambda_k^v, \sigma_k^v, \rho_k^v)$
      set $\delta_{k+1}^f \leftarrow \delta_k^f$ and $\rho_k^f \leftarrow \infty$

procedure COMPUTE_STEPS$(x_k, \delta_k^v, \delta_k^s)$
  set (n_k, λ_k^v) as a primal-dual solution to (3)
  set $(t_k, \lambda_k^f) \leftarrow (0,0)$
  if $\|n_k\| \le \kappa_n \delta_k^s$ and $\|g_k^p\| \ge \kappa_p \|g_k^v\|$ **then**
    set $(t_k, y_k^f, \lambda_k^f)$ as a primal-dual solution to (8)
    if any condition in (14) fails to hold **then** set $(t_k, \lambda_k^f) \leftarrow (0,0)$
  return $(n_k, t_k, \lambda_k^v, \lambda_k^f)$

procedure COMPUTE_SIGMA$(n_k, \lambda_k^v, \sigma_{k-1}^v, \rho_{k-1}^v)$
  if iteration $(k-1)$ was an F-ITERATION **then**
    set $\sigma_k^v \leftarrow \sigma_{k-1}^v$
  else
    if $\rho_{k-1}^v < \kappa_\rho$ **then** set $\sigma_k^v \leftarrow \max\{\sigma_{k-1}^v, \lambda_k^v/\|n_k\|\}$ **else** set $\sigma_k^v \leftarrow \sigma_{k-1}^v$
  return σ_k^v

3. Convergence and complexity analyses for phase 1. The analyses that we present require the following assumption related to the iterate sequence.

Assumption 2. The sequence of iterates $\{x_k\}$ is contained in a compact set. In addition, the sequence $\{\|H_k\|\}$ is bounded over $k \in \mathbb{N}$.

Our analysis makes extensive use of the following mutually exclusive and exhaustive subsets of the iteration index sequence generated by Algorithm 1:

$$\begin{aligned} \mathcal{I} &:= \{k \in \mathbb{N} : \|g_k^v\| > \epsilon\}, \\ \mathcal{F} &:= \{k \in \mathcal{I} : \text{iteration } k \text{ is an F-ITERATION}\}, \\ \text{and} \quad \mathcal{V} &:= \{k \in \mathcal{I} : \text{iteration } k \text{ is a V-ITERATION}\}. \end{aligned}$$

It will also be convenient to define the index set of iterations for which tangential

Algorithm 2 F-ITERATION subroutine

procedure F-ITERATION$(x_k, n_k, s_k, v_k^{\max}, \delta_k^f, \lambda_k^f, \rho_k^f)$
    if $\rho_k^f \geq \kappa_\rho$ **then** [accept step]
        set $x_{k+1} \leftarrow x_k + s_k$
        set $v_{k+1}^{\max}$ according to (17)
        set $\delta_{k+1}^f \leftarrow \max\{\delta_k^f, \gamma_{\mathrm{e}}\|s_k\|\}$
    else (i.e., **if** $\rho_k^f < \kappa_\rho$) [contract trust region]
        set $x_{k+1} \leftarrow x_k$
        set $v_{k+1}^{\max} \leftarrow v_k^{\max}$
        set $\delta_{k+1}^f \leftarrow$ F-CONTRACT$(n_k, s_k, \delta_k^f, \lambda_k^f)$
    return $(x_{k+1}, v_{k+1}^{\max}, \delta_{k+1}^f)$

procedure F-CONTRACT$(n_k, s_k, \delta_k^f, \lambda_k^f)$
    if $\lambda_k^f < \underline{\sigma}\|s_k\|$ **then**
        set $\lambda^f > \lambda_k^f$ so the solution $t(\lambda^f)$ of $\mathcal{Q}_k^f(\lambda^f)$ yields $\underline{\sigma} \leq \lambda^f / \|n_k + t(\lambda^f)\|$
        return $\delta_{k+1}^f \leftarrow \|n_k + t(\lambda^f)\|$
    else (i.e., **if** $\lambda_k^f \geq \underline{\sigma}\|s_k\|$)
        return $\delta_{k+1}^f \leftarrow \gamma_{\mathrm{c}}\|s_k\|$

steps are computed and not reset to zero by our method:

$$\begin{aligned}\mathcal{I}^t &:= \{k \in \mathcal{I} : t_k \neq 0 \text{ when step 7 of Algorithm 1 is reached}\} \\ &= \{k \in \mathcal{I} : \text{step 22 of Algorithm 1 is reached and all conditions in (14) hold}\}.\end{aligned}$$

3.1. Convergence analysis for phase 1. The goal of our convergence analysis is to prove that Algorithm 1 terminates finitely, i.e., $|\mathcal{I}| < \infty$. Our analysis to prove this fact requires a refined examination of the subsets $\mathcal{F}$ and $\mathcal{V}$ of $\mathcal{I}$. For these purposes, we define disjoint subsets of $\mathcal{F}$ as $\mathcal{S}^f := \{k \in \mathcal{F} : \rho_k^f \geq \kappa_\rho\}$ and $\mathcal{C}^f := \{k \in \mathcal{F} : \rho_k^f < \kappa_\rho\}$, and disjoint subsets of $\mathcal{V}$ as

$$\begin{aligned}\mathcal{S}^v &:= \{k \in \mathcal{V} : \rho_k^v \geq \kappa_\rho \text{ and either } \lambda_k^v \leq \sigma_k^v\|n_k\| \text{ or } \|n_k\| = \Delta_k^v\}, \\ \mathcal{C}^v &:= \{k \in \mathcal{V} : \rho_k^v < \kappa_\rho\} \quad \text{and} \quad \mathcal{E}^v := \{k \in \mathcal{V} : k \notin \mathcal{S}^v \cup \mathcal{C}^v\}.\end{aligned}$$

We further partition the set $\mathcal{S}^v$ into the subsets $\mathcal{S}_\Delta^v := \{k \in \mathcal{S}^v : \|n_k\| = \Delta_k^v\}$ and $\mathcal{S}_\sigma^v := \{k \in \mathcal{S}^v : k \notin \mathcal{S}_\Delta^v\}$. Finally, for convenience, we also define the unions $\mathcal{S} := \{k \in \mathcal{I} : k \in \mathcal{S}^f \cup \mathcal{S}^v\}$ and $\mathcal{C} := \{k \in \mathcal{I} : k \in \mathcal{C}^f \cup \mathcal{C}^v\}$. Due to the updates for the primal iterate and/or trust region radii, we refer to iterations with indices in $\mathcal{S}$ as successful, with indices in $\mathcal{C}$ as contractions, and with indices in $\mathcal{E}^v$ as expansions.

Basic relationships between all of these sets are summarized in our first lemma.

LEMMA 3. *The following relationships hold:*
(i) $\mathcal{F} \cap \mathcal{V} = \emptyset$ *and* $\mathcal{F} \cup \mathcal{V} = \mathcal{I}$*;*
(ii) $\mathcal{S}^f \cap \mathcal{C}^f = \emptyset$ *and* $\mathcal{S}^f \cup \mathcal{C}^f = \mathcal{F}$*;*
(iii) $\mathcal{S}^v$, $\mathcal{C}^v$*, and* $\mathcal{E}^v$ *are mutually disjoint and* $\mathcal{S}^v \cup \mathcal{C}^v \cup \mathcal{E}^v = \mathcal{V}$*; and*
(iv) *if* $k \in \mathcal{I} \setminus \mathcal{I}^t$*, then* $k \in \mathcal{V}$*.*

Proof. The fact that $\mathcal{F} \cap \mathcal{V} = \emptyset$ follows from the two cases resulting from the conditional statement in step 9 of Algorithm 1. The rest of parts (i), (ii), and (iii) follow from the definitions of the relevant sets. Part (iv) can be seen to hold as follows. If $k \in \mathcal{I} \setminus \mathcal{I}^t$, then $t_k = 0$ so that (15a) does not hold. It now follows from the logic in Algorithm 1 that $k \in \mathcal{V}$ as claimed. ∎

Algorithm 3 V-ITERATION subroutine

procedure V-ITERATION($x_k, n_k, s_k, v_k^{\max}, \delta_k^v, \Delta_k^v, \lambda_k^v, \sigma_k^v, \rho_k^v$)
    if $\rho_k^v \geq \kappa_\rho$ and either $\lambda_k^v \leq \sigma_k^v \|n_k\|$ or $\|n_k\| = \Delta_k^v$ **then** [accept step]
        set $x_{k+1} \leftarrow x_k + s_k$
        set $v_{k+1}^{\max}$ according to (19)
        set $\Delta_{k+1}^v \leftarrow \max\{\Delta_k^v, \gamma_e \|n_k\|\}$
        set $\delta_{k+1}^v \leftarrow \min\{\Delta_{k+1}^v, \max\{\delta_k^v, \gamma_e \|n_k\|\}\}$
    else if $\rho_k^v < \kappa_\rho$ **then** [contract trust region]
        set $x_{k+1} \leftarrow x_k$
        set $v_{k+1}^{\max} \leftarrow v_k^{\max}$
        set $\Delta_{k+1}^v \leftarrow \Delta_k^v$
        set $\delta_{k+1}^v \leftarrow$ V-CONTRACT($n_k, s_k, \delta_k^v, \lambda_k^v$)
    else (i.e., **if** $\rho_k^v \geq \kappa_\rho$, $\lambda_k^v > \sigma_k^v \|n_k\|$, and $\|n_k\| < \Delta_k^v$) [expand trust region]
        set $x_{k+1} \leftarrow x_k$
        set $v_{k+1}^{\max} \leftarrow v_k^{\max}$
        set $\Delta_{k+1}^v \leftarrow \Delta_k^v$
        set $\delta_{k+1}^v \leftarrow \min\{\Delta_{k+1}^v, \lambda_k^v / \sigma_k^v\}$
    return $(x_{k+1}, v_{k+1}^{\max}, \delta_{k+1}^v, \Delta_{k+1}^v)$

procedure V-CONTRACT($n_k, s_k, \delta_k^v, \lambda_k^v$)
    if $\lambda_k^v < \underline{\sigma} \|n_k\|$ **then**
        set $\hat{\lambda}^v \leftarrow \lambda_k^v + (\underline{\sigma} \|g_k^v\|)^{1/2}$
        set $\lambda^v \leftarrow \hat{\lambda}^v$
        set $n(\lambda^v)$ as the solution of $\mathcal{Q}_k^v(\lambda^v)$
        if $\lambda^v / \|n(\lambda^v)\| \leq \overline{\sigma}$ **then**
            return $\delta_{k+1}^v \leftarrow \|n(\lambda^v)\|$
        else
            set $\lambda^v \in (\lambda_k^v, \hat{\lambda}^v)$ so the solution $n(\lambda^v)$ of $\mathcal{Q}_k^v(\lambda^v)$ yields $\underline{\sigma} \leq \lambda^v / \|n(\lambda^v)\| \leq \overline{\sigma}$
            return $\delta_{k+1}^v \leftarrow \|n(\lambda^v)\|$
    else (i.e., **if** $\lambda_k^v \geq \underline{\sigma} \|n_k\|$)
        set $\lambda^v \leftarrow \gamma_\lambda \lambda_k^v$
        set $n(\lambda^v)$ as the solution of $\mathcal{Q}_k^v(\lambda^v)$
        if $\|n(\lambda^v)\| \geq \gamma_c \|n_k\|$ **then**
            return $\delta_{k+1}^v \leftarrow \|n(\lambda^v)\|$
        else
            return $\delta_{k+1}^v \leftarrow \gamma_c \|n_k\|$

The results in the next lemma are consequences of Assumptions 1 and 2.

LEMMA 4. *The following hold:*
- (i) *there exists* $\theta_{fc} \in (1, \infty)$ *such that* $\max\{\|g_k\|, \|c_k\|, \|J_k\|, \|H_k^v\|\} \leq \theta_{fc}$ *for all* $k \in \mathcal{I}$;
- (ii) $\|g_k^v\| \equiv \|J_k^T c_k\| \leq \theta_{fc} \|c_k\|$ *for all* $k \in \mathcal{I}$; *and*
- (iii) $g^v : \mathbb{R}^N \to \mathbb{R}^N$ *defined by* $g^v(x) = J(x)^T c(x)$ *(recall* (4)*) is Lipschitz continuous with Lipschitz constant* $g_{Lip}^v > 0$ *over an open set containing* $\{x_k\}$.

Proof. Part (i) follows from Assumptions 1 and 2. Part (ii) follows since, by the Cauchy–Schwarz inequality, $\|J_k^T c_k\| \leq \|J_k\| \|c_k\| \leq \theta_{fc} \|c_k\|$. Part (iii) follows since the first derivative of g^v is uniformly bounded under Assumptions 1 and 2. ∎

We now summarize properties associated with the normal and tangential steps.

LEMMA 5. *The following hold for all $k \in \mathcal{I}$:*
(i) $n_k \neq 0$ *and* $s_k \neq 0$*; and*
(ii) *in step 7 of Algorithm 1, the vector t_k satisfies* (14).

Proof. We first prove part (i). Since $k \in \mathcal{I}$, it follows that $\|g_k^v\| > \epsilon$, which combined with (6a) implies that $n_k \neq 0$, as claimed. Now, in order to derive a contradiction, suppose that $0 = s_k = n_k + t_k$, which means that $-t_k = n_k \neq 0$. From $g_k^v \neq 0$ and (6a), it follows that $(H_k^v + \lambda_k^v I)n_k = -g_k^v \neq 0$, which gives

$$n_k^T(H_k^v + \lambda_k^v I)n_k = -n_k^T g_k^v = -n_k^T J_k^T c_k = -(J_k n_k)^T c_k = 0, \tag{20}$$

where the last equality follows from $n_k = -t_k$ and $J_k t_k = 0$ (see (12a)). It now follows from (20), symmetry of $H_k^v + \lambda_k^v I$, and (6b) that $0 = (H_k^v + \lambda_k^v I)n_k = -g_k^v$, which is a contradiction. This completes the proof of part (i).

To prove part (ii), first observe that the conditions in (14) are trivially satisfied if $t_k = 0$. On the other hand, if step 7 is reached with $t_k \neq 0$, then step 22 must have been reached, at which point it must have been determined that all of the conditions in (14) held true (or else t_k would have been reset to the zero vector). ∎

Next, we show that $\{v_k^{\max}\}$ is a monotonically decreasing bound for $\{v_k\}$.

LEMMA 6. *For all $k \in \mathcal{I}$, it follows that $v_k \leq v_k^{\max}$ and $0 < v_{k+1}^{\max} \leq v_k^{\max}$.*

Proof. The result holds trivially if $\mathcal{I} = \emptyset$. Thus, let us assume that $\mathcal{I} \neq \emptyset$, which ensures that $0 \in \mathcal{I}$. Let us now use induction to prove the first inequality, as well as positivity of $v_k^{\max}$ for all $k \in \mathcal{I}$. From the initialization in Algorithm 1, it follows that $v_0 \leq v_0^{\max}$ and $v_0^{\max} > 0$. Now, to complete the induction step, let us assume that $v_k \leq v_k^{\max}$ and $v_k^{\max} > 0$ for some $k \in \mathcal{I}$, then consider three cases.

Case 1 ($k \in \mathcal{S}^f$). When $k \in \mathcal{S}^f$, let us consider the two possibilities based on the procedure for setting $v_{k+1}^{\max}$ stated in (17). If (17) sets $v_{k+1}^{\max} = v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1})$, then the fact that $k \in \mathcal{S}^f \subseteq \mathcal{F}$, (15c), and Lemma 5(i) imply that

$$v_{k+1}^{\max} = v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1}) \geq v_{k+1} + \kappa_{v2}\kappa_\rho\|s_k\|^3 > v_{k+1} \geq 0.$$

On the other hand, if (17) sets $v_{k+1}^{\max} = \max\{\kappa_{v1}v_k^{\max}, v_k^{\max} - \kappa_\rho\|s_k\|^3\}$, then using the induction hypothesis, the fact that $k \in \mathcal{S}^f \subseteq \mathcal{F}$, and (15c), it follows that

$$v_{k+1}^{\max} \geq \kappa_{v1}v_k^{\max} > 0 \quad \text{and} \quad v_{k+1}^{\max} \geq v_k^{\max} - \kappa_\rho\|s_k\|^3 \geq v_{k+1} \geq 0.$$

This case is complete since, in each scenario, $v_{k+1}^{\max} \geq v_{k+1}$ and $v_{k+1}^{\max} > 0$.

Case 2 ($k \in \mathcal{S}^v$). When $k \in \mathcal{S}^v$, let us consider the two possibilities based on the procedure for setting $v_{k+1}^{\max}$ stated in (19). If (19) sets $v_{k+1}^{\max} = v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1})$, then it follows from the induction hypothesis and the fact that $\rho_k^v \geq \kappa_\rho$ for $k \in \mathcal{S}^v$ (which, in particular, implies that $v_{k+1} < v_k$ for $k \in \mathcal{S}^v$) that

$$v_{k+1}^{\max} = v_{k+1} + \kappa_{v2}(v_k^{\max} - v_{k+1}) \geq v_{k+1} + \kappa_{v2}(v_k - v_{k+1}) > v_{k+1} \geq 0.$$

On the other hand, if (19) sets $v_{k+1}^{\max} = \max\{\kappa_{v1}v_k^{\max}, v_{k+1} + \kappa_{v2}(v_k - v_{k+1})\}$, then the induction hypothesis and the fact that $v_{k+1} < v_k$ for $k \in \mathcal{S}^v$ implies that

$$v_{k+1}^{\max} \geq \kappa_{v1}v_k^{\max} > 0 \quad \text{and} \quad v_{k+1}^{\max} \geq v_{k+1} + \kappa_{v2}(v_k - v_{k+1}) > v_{k+1} \geq 0.$$

This case is complete since, in each scenario, $v_{k+1}^{\max} \geq v_{k+1}$ and $v_{k+1}^{\max} > 0$.

Case 3 ($k \notin \mathcal{S}^f \cup \mathcal{S}^v$). When $k \notin \mathcal{S}^f \cup \mathcal{S}^v$, it follows that $k \in \mathcal{C} \cup \mathcal{E}^v$, which may be combined with the induction hypothesis and the updating procedures for x_k and $v_k^{\max}$ in Algorithms 2 and 3 to deduce that $0 < v_k^{\max} = v_{k+1}^{\max}$ and $v_{k+1} = v_k \leq v_k^{\max} = v_{k+1}^{\max}$.

Combining the conclusions of the three cases above, it follows by induction that the first inequality of the lemma holds true and $v_k^{\max} > 0$ for all $k \in \mathcal{I}$.

Let us now prove that $v_{k+1}^{\max} \leq v_k^{\max}$ for all $k \in \mathcal{I}$, again by considering three cases. First, if $k \in \mathcal{S}^f$, then $v_{k+1}^{\max}$ is set using (17) such that

$$v_{k+1}^{\max} \leq \max\{\kappa_{v1} v_k^{\max}, v_k^{\max} - \kappa_\rho \|s_k\|^3\} < v_k^{\max},$$

where the strict inequality follows by $\kappa_{v1} \in (0,1)$ and Lemma 5(i). Second, if $k \in \mathcal{S}^v$, then $v_{k+1} < v_k \leq v_k^{\max}$, where we have used the proved fact that $v_k \leq v_k^{\max}$; thus, $v_k^{\max} - v_{k+1} > 0$. Then, since $v_{k+1}^{\max}$ is set using (19), it follows that

$$v_k^{\max} - v_{k+1}^{\max} \geq v_k^{\max} - v_{k+1} - \kappa_{v2}(v_k^{\max} - v_{k+1}) = (1 - \kappa_{v2})(v_k^{\max} - v_{k+1}) > 0.$$

Third, if $k \notin \mathcal{S}^f \cup \mathcal{S}^v$, then, by construction in Algorithms 2 and 3, $v_{k+1}^{\max} = v_k^{\max}$. ☐

Our next lemma gives a lower bound for the decrease in the trust funnel radius.

LEMMA 7. *If* $k \in \mathcal{S}$, *then* $v_k^{\max} - v_{k+1}^{\max} \geq \kappa_\rho(1 - \kappa_{v2})\|s_k\|^3$.

Proof. If $k \in \mathcal{S}^f$, then $v_{k+1}^{\max}$ is set using (17). In this case,

$$\begin{aligned} v_k^{\max} - v_{k+1}^{\max} &\geq v_k^{\max} - v_{k+1} - \kappa_{v2}(v_k^{\max} - v_{k+1}) \\ &= (1 - \kappa_{v2})(v_k^{\max} - v_{k+1}) \geq \kappa_\rho(1 - \kappa_{v2})\|s_k\|^3, \end{aligned}$$

where the last inequality follows from (15c) (since $k \in \mathcal{S}^f \subseteq \mathcal{F}$). If $k \in \mathcal{S}^v$, then $v_{k+1}^{\max}$ is set using (19). In this case, by Lemma 6 and the fact that $\rho_k^v \geq \kappa_\rho$ for $k \in \mathcal{S}^v$,

$$\begin{aligned} v_k^{\max} - v_{k+1}^{\max} &\geq v_k^{\max} - v_{k+1} - \kappa_{v2}(v_k^{\max} - v_{k+1}) \\ &= (1 - \kappa_{v2})(v_k^{\max} - v_{k+1}) \geq (1 - \kappa_{v2})(v_k - v_{k+1}) \geq \kappa_\rho(1 - \kappa_{v2})\|s_k\|^3, \end{aligned}$$

which completes the proof. ☐

Subsequently in our analysis, it will be convenient to consider an alternative formulation of problem (8) that arises from an orthogonal decomposition of the normal step n_k into its projection onto the range space of J_k^T, call it n_k^R, and its projection onto the null space of J_k, call it n_k^N. Specifically, considering

$$(21) \quad t_k^N \in \arg\min_{t^N \in \mathbb{R}^N} m_k^f(n_k^R + t^N) \quad \text{s.t.} \quad J_k t^N = 0 \quad \text{and} \quad \|t^N\| \leq \sqrt{(\delta_k^s)^2 - \|n_k^R\|^2},$$

we can recover the solution of (8) as $t_k \leftarrow t_k^N - n_k^N$. Similarly, for any $\lambda_k^f \in [0, \infty)$ that is strictly greater than the leftmost eigenvalue of $Z_k^T H_k Z_k$, let us define

$$(22) \quad \bar{\mathcal{Q}}_k^f(\lambda_k^f): \quad \min_{t^N \in \mathbb{R}^N} (g_k + H_k n_k^R)^T t^N + \tfrac{1}{2}(t^N)^T(H_k + \lambda_k^f I)t^N \quad \text{s.t.} \quad J_k t^N = 0.$$

In the next lemma, we formally establish the equivalence between problems (21) and (8), as well as between problems (22) and (13).

LEMMA 8. *For all* $k \in \mathcal{I}$, *the following problem equivalences hold:*

(i) *if* $\|n_k\| \leq \delta_k^s$, *then problems* (21) *and* (8) *are equivalent in that* (t_k^N, λ_k^N) *is part of a primal-dual solution of problem* (21) *if and only if* $(t_k, \lambda_k^f) = (t_k^N - n_k^N, \lambda_k^N)$ *is part of a primal-dual solution of problem* (8); *and*

(ii) *if* $Z_k^T H_k Z_k + \lambda_k^f I \succ 0$*, then problems* (22) *and* (13) *are equivalent in that* t_k^N *solves problem* (22) *if and only if* $t_k = t_k^N - n_k^N$ *solves problem* (13)*.*

Proof. To prove part (i), first note that $\|n_k\| \le \delta_k^s$ ensures that problems (21) and (8) are feasible. Then, by $J_k t^N = 0$ in (21), the vector $n_k^R \in \text{Range}(J_k^T)$ is orthogonal with any feasible solution of (21), meaning that the trust region constraint in (21) is equivalent to $\|n_k^R + t^N\| \le \delta_k^s$. Thus, as (12) are the optimality conditions of (8), the optimality conditions of problem (21) (with this modified trust region constraint) are that there exists $(t_k^N, y_k^N, \lambda_k^N) \in \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}$ such that

$$\begin{bmatrix} H_k + \lambda_k^N I & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} t_k^N \\ y_k^N \end{bmatrix} = - \begin{bmatrix} g_k + (H_k + \lambda_k^N I) n_k^R \\ 0 \end{bmatrix}, \tag{23a}$$

$$Z_k^T H_k Z_k + \lambda_k^N I \succeq 0, \tag{23b}$$

$$\text{and} \quad \lambda_k^N \perp (\delta_k^s - \|n_k^R + t_k^N\|) \ge 0. \tag{23c}$$

From equivalence of the systems (23) and (12), it is clear that $(t_k^N, y_k^N, \lambda_k^N)$ is a primal-dual solution of (21) (with the modified trust region constraint) if and only if $(t_k, y_k^f, \lambda_k^f) = (t_k^N - n_k^N, y_k^N, \lambda_k^N)$ is a primal-dual solution of (8). This proves part (i). Part (ii) follows in a similar manner from the orthogonal decomposition $n_k = n_k^N + n_k^R$, and the fact that $J_k t^N = 0$ in (22) ensures that $t_k^N \in \text{Null}(J_k)$. ◻

The next lemma reveals important properties of the tangential step. In particular, it shows that the procedure for performing a contraction of the trust region radius in an F-ITERATION that results in a rejected step is well defined.

LEMMA 9. *If* $k \in \mathcal{C}^f$ *and the condition in step* 12 *of Algorithm* 2 *tests true, then there exists* $\lambda^f > \lambda_k^f$ *such that* $\underline{\sigma} \le \lambda^f / \|n_k + t(\lambda^f)\|$*, where* $t(\lambda^f)$ *solves* $\mathcal{Q}_k^f(\lambda^f)$*.*

Proof. Since the condition in step 12 of Algorithm 2 is assumed to test true, it follows that $\lambda_k^f < \underline{\sigma}\|s_k\|$. Second, letting $t(\lambda^f)$ denote the solution of $\mathcal{Q}_k^f(\lambda^f)$, it follows by Lemma 8(ii) that $\lim_{\lambda^f \to \infty} \|n_k + t(\lambda^f)\| = \|n_k^R\|$, meaning that $\lim_{\lambda^f \to \infty} \lambda^f / \|n_k + t(\lambda^f)\| = \infty$. It follows from these observations and standard theory for trust region methods [12, Chapter 7] that the result is true. ◻

The next lemma reveals properties of the normal step trust region radii along with some additional observations about the sequences $\{\Delta_k^v\}$, $\{\lambda_k^v\}$, and $\{\sigma_k^v\}$.

LEMMA 10. *The following hold:*

(i) *if* $k \in \mathcal{C}^v$*, then* $0 < \delta_{k+1}^v < \delta_k^v$ *and* $\lambda_{k+1}^v \ge \lambda_k^v$*;*
(ii) *if* $k \in \mathcal{I}$*, then* $\delta_k^v \le \Delta_k^v \le \Delta_{k+1}^v$*;*
(iii) *if* $k \in \mathcal{S}^v \cup \mathcal{E}^v$*, then* $\delta_{k+1}^v \ge \delta_k^v$*; and*
(iv) *if* $k \in \mathcal{F}$*, then* $\delta_{k+1}^v = \delta_k^v$ *and* $\sigma_{k+1}^v = \sigma_k^v$*.*

Proof. The proof of part (i) follows in the same way as that of [17, Lemma 3.4]. In particular, since the V-CONTRACT procedure follows exactly that of CONTRACT in [17], it follows that any call of V-CONTRACT results in a contraction of the trust region radius for the normal subproblem and nondecrease of the corresponding dual variable.

For part (ii), the result is trivial if $\mathcal{I} = \emptyset$. Thus, let us assume that $\mathcal{I} \ne \emptyset$, which ensures that $0 \in \mathcal{I}$. We now first prove $\delta_k^v \le \Delta_k^v$ for all $k \in \mathcal{I}$ using induction. By the initialization procedure of Algorithm 1, it follows that $\delta_0^v \le \Delta_0^v$. Hence, let us proceed by assuming that $\delta_k^v \le \Delta_k^v$ for some $k \in \mathcal{I}$. If $k \in \mathcal{S}^v$, then step 6 of Algorithm 3 shows that $\delta_{k+1}^v \le \Delta_{k+1}^v$. If $k \in \mathcal{E}^v$, then step 16 of Algorithm 3 gives $\delta_{k+1}^v \le \Delta_{k+1}^v$. If $k \in \mathcal{C}^v$, then part (i), step 10 of Algorithm 3, and the induction hypothesis yield

$\delta^v_{k+1} < \delta^v_k \le \Delta^v_k = \Delta^v_{k+1}$. Lastly, if $k \in \mathcal{F}$, then step 12 of Algorithm 1 and the inductive hypothesis give $\delta^v_{k+1} = \delta^v_k \le \Delta^v_k = \Delta^v_{k+1}$. The induction step has now been completed since we have overall proved that $\delta^v_{k+1} \le \Delta^v_{k+1}$, which means that we have proved the first inequality in part (ii). To prove $\Delta^v_k \le \Delta^v_{k+1}$, consider two cases. If $k \in \mathcal{S}^v$, then step 5 of Algorithm 3 gives $\Delta^v_{k+1} \ge \Delta^v_k$. Otherwise, if $k \notin \mathcal{S}^v$, then according to step 12 of Algorithm 1 and steps 10 and 15 of Algorithm 3, it follows that $\Delta^v_{k+1} = \Delta^v_k$. Combining both cases, the proof of part (ii) is now complete.

For part (iii), first observe from part (ii) and step 6 of Algorithm 3 that if $k \in \mathcal{S}^v$, then $\delta^v_{k+1} = \min\{\Delta^v_{k+1}, \max\{\delta^v_k, \gamma_e\|n_k\|\}\} \ge \delta^v_k$. On the other hand, if $k \in \mathcal{E}^v$, then the conditions that must hold true for step 12 of Algorithm 3 to be reached ensure that $\lambda^v_k > 0$, meaning that $\|n_k\| = \delta^v_k$ (see (6c)). From this and the fact that the conditions in step 12 of Algorithm 3 must hold true, it follows that $\lambda^v_k/\sigma^v_k > \|n_k\| = \delta^v_k$ and $\|n_k\| < \Delta^v_k$. Combining these observations with $\Delta^v_{k+1} = \Delta^v_k$ for $k \in \mathcal{E}^v$ (see step 15 of Algorithm 3) it follows from step 16 of Algorithm 3 that $\delta^v_{k+1} > \|n_k\| = \delta^v_k$.

Finally, part (iv) follows from steps 12 and 26 of Algorithm 1. ∎

The next result reveals similar properties for the other radii and $\{\lambda^f_k\}$.

LEMMA 11. *The following hold:*

(i) *if $k \in \mathcal{C}^f$, then $\delta^f_{k+1} < \delta^f_k$ and if, in addition, $(k+1) \in \mathcal{I}^t$, then $\lambda^f_{k+1} \ge \lambda^f_k$;*

(ii) *if $k \in \mathcal{S}^f$, then $\delta^f_{k+1} \ge \delta^f_k$ and $\delta^s_{k+1} \ge \delta^s_k$; and*

(iii) *if $k \in \mathcal{V}$, then $\delta^f_{k+1} = \delta^f_k$.*

Proof. For part (i), notice that δ^f_{k+1} is set in step 9 of Algorithm 2 and that $(x_{k+1}, \delta^v_{k+1}) \leftarrow (x_k, \delta^v_k)$ and $n_{k+1} = n_k$ for all $k \in \mathcal{C}^f$. Let us proceed by considering two cases depending on the condition in step 12 of Algorithm 2.

Case 1 ($\lambda^f_k < \underline{\sigma}\|s_k\|$). In this case, δ^f_{k+1} is set in step 14 of Algorithm 2, which from step 13 of Algorithm 2 and Lemma 9 implies that $\lambda^f > \lambda^f_k$. Combining this with Lemma 8 and standard theory for trust region methods leads to the fact that the solution $t^N(\lambda^f)$ of $\bar{\mathcal{Q}}^f_k(\lambda^f)$ satisfies $\|t^N(\lambda^f)\| < \|t^N_k\|$. Thus, $\delta^f_{k+1} = \|n_k + t(\lambda^f)\| = \|n^R_k + t^N(\lambda^f)\| < \|n^R_k + t^N_k\| = \|s_k\| \le \delta^f_k$, where the last inequality comes from (10). If, in addition, $(k+1) \in \mathcal{I}^t$ so that a nonzero tangential step is computed and not reset to zero, it follows that $\lambda^f_{k+1} = \lambda^f$. This establishes the last conclusion of part (i) for this case since it has already been shown above that $\lambda^f > \lambda^f_k$.

Case 2 ($\lambda^f_k \ge \underline{\sigma}\|s_k\|$). In this case, δ^f_{k+1} is set in step 16 of Algorithm 2 and, from (10) and $\gamma_c \in (0,1)$, it follows that $\delta^f_{k+1} = \gamma_c\|s_k\| \le \gamma_c\delta^f_k < \delta^f_k$. Consequently, from step 12 of Algorithm 1 and (10), one finds that $\delta^s_{k+1} \le \delta^s_k$. It then follows from Lemma 8 and standard trust region theory that if $(k+1) \in \mathcal{I}^t$, then $\lambda^f_{k+1} \ge \lambda^f_k$.

To prove part (ii), notice that for $k \in \mathcal{S}^f$ it follows by step 5 of Algorithm 2 that $\delta^f_{k+1} = \max\{\delta^f_k, \gamma_e\|s_k\|\}$, so $\delta^f_{k+1} \ge \delta^f_k$. From this, step 12 of Algorithm 1, and (10) it follows that $\delta^s_{k+1} \ge \delta^s_k$. These conclusions complete the proof of part (ii).

Finally, part (iii) follows from step 16 of Algorithm 1. ∎

Next, we show that after a V-ITERATION with either a contraction or an expansion of the trust region radius, the subsequent iteration cannot result in an expansion.

LEMMA 12. *If $k \in \mathcal{C}^v \cup \mathcal{E}^v$, then $(k+1) \in \mathcal{F} \cup \mathcal{S}^v \cup \mathcal{C}^v$.*

Proof. If $(k+1) \in \mathcal{F}$, then there is nothing left to prove. Otherwise, if $(k+1) \in \mathcal{V}$, then the proof follows using the same logic as for [17, Lemma 3.7], which shows that one of three cases holds: (i) $k \in \mathcal{C}^v$, which yields $\lambda^v_{k+1} \le \sigma^v_{k+1}\|n_{k+1}\|$, so $(k+1) \notin \mathcal{E}^v$;

(ii) $k \in \mathcal{E}^v$ and $\Delta_k^v \geq \lambda_k^v/\sigma_k^v$, which also yields $\lambda_{k+1}^v \leq \sigma_{k+1}^v \|n_{k+1}\|$, so $(k+1) \notin \mathcal{E}^v$; or (iii) $k \in \mathcal{E}^v$ and $\Delta_k^v < \lambda_k^v/\sigma_k^v$, which implies $(k+1) \in \mathcal{S}^v \cup \mathcal{C}^v$, so $(k+1) \notin \mathcal{E}^v$. ◻

Our goal now is to expand upon the conclusions of Lemma 12. To do this, it will be convenient to define the first index in a given index set following an earlier index $\bar{k} \in \mathcal{I}$ in that index set (or the initial index 0). In particular, let us define

$$k_{\mathcal{S}}(\bar{k}) := \min\{k \in \mathcal{S} : k > \bar{k}\} \quad \text{and} \quad k_{\mathcal{S}\cup\mathcal{V}}(\bar{k}) := \min\{k \in \mathcal{S} \cup \mathcal{V} : k > \bar{k}\}$$

along with the associated sets

$$\mathcal{I}_{\mathcal{S}}(\bar{k}) := \{k \in \mathcal{I} : \bar{k} < k < k_{\mathcal{S}}(\bar{k})\} \quad \text{and} \quad \mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k}) := \{k \in \mathcal{I} : \bar{k} < k < k_{\mathcal{S}\cup\mathcal{V}}(\bar{k})\}.$$

The following lemma shows one important property related to these quantities.

LEMMA 13. *For all $\bar{k} \in \mathcal{S} \cup \{0\}$, it follows that $|\mathcal{E}^v \cap \mathcal{I}_{\mathcal{S}}(\bar{k})| \leq 1$.*

Proof. In order to derive a contradiction, suppose that there exists $\bar{k} \in \mathcal{S} \cup \{0\}$ such that $|\mathcal{E}^v \cap \mathcal{I}_{\mathcal{S}}(\bar{k})| > 1$, which means that one can choose $k_{\mathcal{S}_1}$ and $k_{\mathcal{S}_3}$ as the first two distinct indices in $\mathcal{E}^v \cap \mathcal{I}_{\mathcal{S}}(\bar{k})$; in particular,

$$\{k_{\mathcal{S}_1}, k_{\mathcal{S}_3}\} \subseteq \mathcal{E}^v \cap \mathcal{I}_{\mathcal{S}}(\bar{k}) \quad \text{and} \quad \bar{k} < k_{\mathcal{S}_1} < k_{\mathcal{S}_3} < k_{\mathcal{S}}(\bar{k}).$$

By Lemma 12 and the fact that $k_{\mathcal{S}_1} \in \mathcal{E}^v$, it follows that $\{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\} \neq \emptyset$. Let us proceed by considering two cases, deriving a contradiction in each case.

Case 1 ($\mathcal{V} \cap \{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\} = \emptyset$). In this case, by the definitions of $k_{\mathcal{S}_1}$, $k_{\mathcal{S}_3}$, and $\mathcal{I}_{\mathcal{S}}(\bar{k})$, it follows that $\{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\} \subseteq \mathcal{C}^f$. Then, since $\delta_{k+1}^v = \delta_k^v$ and $\sigma_{k+1}^v = \sigma_k^v$ for all $k \in \mathcal{C}^f \subseteq \mathcal{F}$, it follows that $\delta_{k_{\mathcal{S}_3}}^v = \delta_{k_{\mathcal{S}_1}+1}^v$ and $\sigma_{k_{\mathcal{S}_3}}^v = \sigma_{k_{\mathcal{S}_1}+1}^v$. In particular, using the fact that $\delta_{k_{\mathcal{S}_3}}^v = \delta_{k_{\mathcal{S}_1}+1}^v$, it follows along with the fact that $x_{k+1} = x_k$ for all $k \notin \mathcal{S}$ that $\|n_{k_{\mathcal{S}_3}}\| = \|n_{k_{\mathcal{S}_1}+1}\|$ and $\lambda_{k_{\mathcal{S}_3}}^v = \lambda_{k_{\mathcal{S}_1}+1}^v$. Now, since $(k_{\mathcal{S}_1}+1) \in \mathcal{C}^f$, it follows with step 9 of Algorithm 1 and (15e) that

$$\lambda_{k_{\mathcal{S}_3}}^v/\|n_{k_{\mathcal{S}_3}}\| = \lambda_{k_{\mathcal{S}_1}+1}^v/\|n_{k_{\mathcal{S}_1}+1}\| \leq \sigma_{k_{\mathcal{S}_1}+1}^v = \sigma_{k_{\mathcal{S}_3}}^v,$$

which implies that $k_{\mathcal{S}_3} \notin \mathcal{E}^v$, a contradiction.

Case 2 ($\mathcal{V} \cap \{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\} \neq \emptyset$). In this case, by the definitions of $k_{\mathcal{S}_1}$, $k_{\mathcal{S}_3}$, and $\mathcal{I}_{\mathcal{S}}(\bar{k})$, it follows that $\{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\} \subseteq \mathcal{C}^f \cup \mathcal{C}^v$. In addition, by the condition of this case, it also follows that there exists a greatest index $k_{\mathcal{S}_2} \in \mathcal{C}^v \cap \{k_{\mathcal{S}_1}+1, \ldots, k_{\mathcal{S}_3}-1\}$. In particular, for the index $k_{\mathcal{S}_2} \in \mathcal{C}^v$, it follows that $k_{\mathcal{S}_1}+1 \leq k_{\mathcal{S}_2} \leq k_{\mathcal{S}_3}-1$ and $\{k_{\mathcal{S}_2}+1, \ldots, k_{\mathcal{S}_3}-1\} \subseteq \mathcal{C}^f$. By $k_{\mathcal{S}_2} \in \mathcal{C}^v$ and Lemma 12, it follows that $k_{\mathcal{S}_2}+1 \notin \mathcal{E}^v$; hence, since $k_{\mathcal{S}_3} \in \mathcal{E}^v$, it follows that $\{k_{\mathcal{S}_2}+1, \ldots, k_{\mathcal{S}_3}-1\} \neq \emptyset$. We may now apply the same argument as for Case 1, but with $k_{\mathcal{S}_1}$ replaced by $k_{\mathcal{S}_2}$, to arrive at the contradictory conclusion that $k_{\mathcal{S}_3} \notin \mathcal{E}^v$, completing the proof.

The next lemma reveals lower bounds for the norms of the normal and full steps.

LEMMA 14. *For all $k \in \mathcal{I}$, the following hold:*
(i) $\|n_k\| \geq \min\{\delta_k^v, \|g_k^v\|/\|H_k^v\|\} > 0$ *and*
(ii) $\|s_k\| \geq \kappa_{ntn} \min\{\delta_k^v, \|g_k^v\|/\|H_k^v\|\} > 0$.

Proof. The proof of part (i) follows in the same way as that for [17, Lemma 3.2]. Part (ii) follows from part (i) and (14b), the latter of which holds by Lemma 5(ii). ◻

We now provide a lower bound for the decrease in the model of infeasibility.

LEMMA 15. *For all $k \in \mathcal{I}$, the quantities n_k, λ_k^v, and s_k satisfy*

$$v_k - m_k^v(n_k) = \tfrac{1}{2} n_k^T (H_k^v + \lambda_k^v I) n_k + \tfrac{1}{2}\lambda_k^v \|n_k\|^2 > 0, \tag{24a}$$

$$v_k - m_k^v(s_k) \geq \kappa_{vm}(\tfrac{1}{2} n_k^T (H_k^v + \lambda_k^v I) n_k + \tfrac{1}{2}\lambda_k^v \|n_k\|^2) > 0, \quad \textit{and} \tag{24b}$$

$$v_k - m_k^v(s_k) \geq \tfrac{1}{2}\kappa_{vm}\|g_k^v\| \min\{\delta_k^v, \|g_k^v\|/\|H_k^v\|\} > 0. \tag{24c}$$

Proof. The proof of (24a) follows in the same way as for that of [17, Lemma 3.3] and from the fact that $\|g_k^v\| > \varepsilon$, which holds since $k \in \mathcal{I}$. The inequalities in (24b) follow from (24a) and (14a), the latter of which holds because of Lemma 5(ii). To prove (24c), first observe from standard trust region theory (e.g., see [12, Theorem 6.3.1]) that

$$v_k - m_k^v(n_k) \geq \tfrac{1}{2}\|g_k^v\| \min\{\delta_k^v, \|g_k^v\|/\|H_k^v\|\} > 0. \tag{25}$$

By combining (25) and (14a) (which holds by Lemma 5(ii)), one obtains (24c). ◻

The next lemma reveals that if the dual variable for the normal step trust region is beyond a certain threshold, then the trust region constraint must be active and the step will either be an F-ITERATION or a successful V-ITERATION.

LEMMA 16. *For all $k \in \mathcal{I}$, if the trial step s_k and the dual variable λ_k^v satisfy*

$$\lambda_k^v \geq \kappa_\delta^2(2g_{Lip}^v + \theta_{fc} + 2\kappa_\rho\|s_k\|)/\kappa_{vm}, \tag{26}$$

then $\|n_k\| = \delta_k^v$ and $\rho_k^v \geq \kappa_\rho$.

Proof. For all $k \in \mathcal{I}$, it follows from the definition of m_k^v and the mean value theorem that there exists a point $\bar{x}_k \in \mathbb{R}^N$ on the line segment $[x_k, x_k + s_k]$ such that

$$\begin{aligned} m_k^v(s_k) - v(x_k + s_k) &= (g_k^v - g^v(\bar{x}_k))^T s_k + \tfrac{1}{2} s_k^T H_k^v s_k \\ &\geq -\|g_k^v - g^v(\bar{x}_k)\|\|s_k\| - \tfrac{1}{2}\|H_k^v\|\|s_k\|^2. \end{aligned} \tag{27}$$

By (26) and (6c), it follows that $\|n_k\| = \delta_k^v$. Combining this fact with (27), (24b), (6b), Lemma 4, (26), and the fact that $\|s_k\| \leq \delta_k^s \leq \kappa_\delta \delta_k^v = \kappa_\delta\|n_k\|$, one obtains

$$\begin{aligned} v_k - v(x_k + s_k) &= v_k - m_k^v(s_k) + m_k^v(s_k) - v(x_k + s_k) \\ &\geq \tfrac{1}{2}\kappa_{vm}\lambda_k^v\|n_k\|^2 - \|g_k^v - g^v(\bar{x}_k)\|\|s_k\| - \tfrac{1}{2}\|H_k^v\|\|s_k\|^2 \\ &\geq (\tfrac{1}{2}\kappa_{vm}\kappa_\delta^{-2}\lambda_k^v - g_{Lip}^v - \tfrac{1}{2}\theta_{fc})\|s_k\|^2 \geq \kappa_\rho\|s_k\|^3, \end{aligned}$$

which, by steps 12 and 14 of Algorithm 1 and (18), completes the proof. ◻

Recall that our main goal in this section is to prove that $|\mathcal{I}| < \infty$. Ultimately, this result is attained by deriving contradictions under the assumption that $|\mathcal{I}| = \infty$. For example, if $|\mathcal{I}| = \infty$ and the iterations corresponding to all sufficiently large $k \in \mathcal{I}$ involve contractions of a trust region radius, then the following lemma helps to lead to contradictions in subsequent results. In particular, it reveals that, under these conditions, a corresponding dual variable tends to infinity.

LEMMA 17. *The following hold.*

(i) *If $k \notin \mathcal{S}$ for all large $k \in \mathcal{I}$ and $|\mathcal{C}^v| = \infty$, then $\{\delta_k^v\} \to 0$ and $\{\lambda_k^v\} \to \infty$.*

(ii) *If $k \in \mathcal{C}^f$ for all large $k \in \mathcal{I}$, then $\{\delta_k^f\} \to 0$ and $\{\lambda_k^f\} \to \infty$.*

Proof. By Lemmas 10 and 13, and the fact that $k \notin \mathcal{S}$ for all large $k \in \mathcal{I}$, the proof of part (i) follows in the same way as that of [17, Lemma 3.9].

To prove part (ii), let us assume, without loss of generality, that $k \in \mathcal{C}^f$ for all $k \in \mathcal{I}$. It then follows that $k \in \mathcal{I}^t$ for all $k \in \mathcal{I}$, since otherwise it would follow that $t_k \leftarrow 0$, which by (15a) means $k \in \mathcal{V}$, a contradiction to $k \in \mathcal{C}^f$. Thus,

$$k \in \mathcal{C}^f \cap \mathcal{I}^t \quad \text{for all} \quad k \in \mathcal{I}. \tag{28}$$

Next, we claim that the condition in step 12 of Algorithm 2 can hold true for at most one iteration. If it never holds true, then there is nothing left to prove. Otherwise, let $k_c \in \mathcal{I}$ be the first index for which the condition holds true. The structure of Algorithm 2 (see step 13) and (28) then ensure that $\lambda^f_{k_c+1}/\|s_{k_c+1}\| \geq \underline{\sigma}$. From Lemma 11(i), one may conclude that $\{\lambda^f_k/\|s_k\|\}$ is nondecreasing. From this, it follows that the condition in step 12 of Algorithm 2 will never be true for any $k > k_c$. Thus, we may now proceed, without loss of generality, under the assumption that the condition in step 12 of Algorithm 2 always tests false. This means that δ^f_{k+1} is set in step 16 of Algorithm 2 for all $k \in \mathcal{I}$, yielding $\delta^f_{k+1} \leftarrow \gamma_c\|s_k\| \leq \gamma_c\delta^f_k$, where the last inequality comes from (10). Therefore, $\{\delta^f_k\} \to 0$ for all $k \in \mathcal{I}$, and consequently $\{\lambda^f_k\} \to \infty$. ◻

We now show that the sequences $\{\Delta^v_k\}$ and $\{n_k\}$ are bounded above.

LEMMA 18. *There exists a scalar $\Delta^v_{\max} \in (0,\infty)$ such that $\Delta^v_k = \Delta^v_{\max}$ for all sufficiently large $k \in \mathcal{I}$. In addition, $|\mathcal{S}^v_\Delta| < \infty$ and there exists a scalar $n_{\max} \in (0,\infty)$ such that $\|n_k\| \leq n_{\max}$ for all $k \in \mathcal{I}$.*

Proof. First, in order to derive a contradiction, assume that there is no $\Delta^v_{\max}$ such that $\Delta^v_k = \Delta^v_{\max}$ for all sufficiently large $k \in \mathcal{I}$. This, in turn, means that step 5 of Algorithm 3 is reached infinitely often, meaning that $|\mathcal{S}^v| = \infty$. For all $k \in \mathcal{S}^v \subseteq \mathcal{S}$, it follows from Lemma 7 that $v^{\max}_k - v^{\max}_{k+1} \geq \kappa_\rho(1-\kappa_{v2})\|s_k\|^3$. Now, using the monotonicity of $\{v^{\max}_k\}$ and the fact that $v^{\max}_k \geq 0$ (see Lemma 6), one may conclude that $\{v^{\max}_k\}$ converges; therefore $\{s_k\}_{k\in\mathcal{S}^v} \to 0$. From this fact, Lemma 5(ii), and (14b) it follows that $\{n_k\}_{k\in\mathcal{S}^v} \to 0$. Thus, there exists an iteration index k^v_Δ such that for all $k \in \mathcal{S}^v$ with $k \geq k^v_\Delta$, one finds $\gamma_e\|n_k\| < \Delta^v_0 \leq \Delta^v_k$, where the last inequality follows from Lemma 10(ii). From this and steps 5, 10, and 15 of Algorithm 3, it follows that $\Delta^v_{k+1} \leftarrow \Delta^v_k$ for all $k \geq k^v_\Delta$, a contradiction. The proof of the second part of the lemma follows in the same way as in that for [17, Lemma 3.11]. ◻

In the next lemma, a uniform lower bound on $\{\delta^v_k\}$ is provided.

LEMMA 19. *There exists a scalar $\delta^v_{\min} \in (0,\infty)$ such that $\delta^v_k \geq \delta^v_{\min}$ for all $k \in \mathcal{I}$.*

Proof. If $|\mathcal{C}^v| < \infty$, then the result follows from Lemma 10(iii)–(iv). Thus, let us proceed under the assumption that $|\mathcal{C}^v| = \infty$. As in the beginning of the proof of Lemma 16, it follows that (27) holds. Then, using (27), (24c), Lemmas 4(i) and 4(iii), $\|g^v_k\| > \epsilon$ for $k \in \mathcal{I}$, and $\|s_k\| \leq \delta^s_k \leq \kappa_\delta\delta^v_k$, it follows that

$$\begin{aligned} v_k - v(x_k+s_k) &= v_k - m^v_k(s_k) + m^v_k(s_k) - v(x_k+s_k) \\ &\geq \tfrac{1}{2}\kappa_{vm}\epsilon\min\{\delta^v_k, \epsilon/\theta_{fc}\} - (g^v_{Lip} + \tfrac{1}{2}\theta_{fc})\kappa^2_\delta(\delta^v_k)^2. \end{aligned}$$

Considering these inequalities and $\|s_k\| \leq \delta^s_k \leq \kappa_\delta\delta^v_k$, it must hold that $\rho^v_k \geq \kappa_\rho$ for any $k \in \mathcal{I}$ as long as $\delta^v_k \in (0, \epsilon/\theta_{fc}]$ is sufficiently small such that

$$\tfrac{1}{2}\kappa_{vm}\epsilon\delta^v_k - (g^v_{Lip} + \tfrac{1}{2}\theta_{fc})\kappa^2_\delta(\delta^v_k)^2 \geq \kappa_\rho\kappa^3_\delta(\delta^v_k)^3 \geq \kappa_\rho\|s_k\|^3.$$

This fact implies the existence of a positive threshold $\delta^v_{thresh} \in (0, \epsilon/\theta_{fc}]$ such that, for any $k \in \mathcal{I}$ with $\delta^v_k \in (0, \delta^v_{thresh})$, one finds $\rho^v_k \geq \kappa_\rho$. Along with the fact that

$\rho_k^v < \kappa_\rho$ if and only if $k \in \mathcal{C}^v$ (see steps 2, 7, and 12 of Algorithm 3 and step 12 of Algorithm 1), it follows that

$$\delta_k^v \geq \delta_{thresh}^v \quad \text{for all } k \in \mathcal{C}^v. \tag{29}$$

Since the normal step subproblem trust region radius is only decreased when $k \in \mathcal{C}^v$, we will complete the proof by showing a lower bound on δ_{k+1}^v when $k \in \mathcal{C}^v$.

Suppose that $k \in \mathcal{C}^v$. If step 24 of Algorithm 3 is reached, then

$$\delta_{k+1}^v \leftarrow \|n(\lambda^v)\| \geq \lambda^v/\overline{\sigma} = (\lambda_k^v + (\underline{\sigma}\|g_k^v\|)^{1/2})/\overline{\sigma} \geq (\underline{\sigma}\|g_k^v\|)^{1/2}/\overline{\sigma} \geq (\underline{\sigma}\epsilon)^{1/2}/\overline{\sigma},$$

where the last inequality follows since $k \in \mathcal{I}$ means $\|g_k^v\| \geq \epsilon$. If step 27 is reached, then the algorithm chooses $\lambda^v \in (\lambda_k^v, \hat{\lambda}^v)$ to find an $n(\lambda^v)$ that solves $\mathcal{Q}_k^v(\lambda^v)$ such that $\underline{\sigma} \leq \lambda^v/\|n(\lambda^v)\| \leq \overline{\sigma}$. For this case and the cases when step 32 or 34 is reached, the existence of $\delta_{\min}^v \in (0,\infty)$ such that $\delta_{k+1}^v \geq \delta_{\min}^v$ for all $k \in \mathcal{C}^v$ follows in the same manner as in the proof of [17, Lemma 3.12]. Combining these facts with (29) and Lemma 10(iii)–(iv), the proof is complete. ∎

The next result shows that there are finitely many successful iterations.

LEMMA 20. *The following hold:* $|\mathcal{S}^v| < \infty$ *and* $|\mathcal{S}^f| < \infty$.

Proof. Lemma 19, $\|g_k^v\| > \epsilon$ for all $k \in \mathcal{I}$, Lemmas 14(i) and 4(i) imply the existence of $n_{\min} \in (0,\infty)$ such that $\|n_k\| \geq n_{\min}$ for all $k \in \mathcal{I}$, i.e.,

$$\|g_k^v\| > \epsilon \quad \text{and} \quad \|n_k\| \geq n_{\min} > 0 \quad \text{for all } k \in \mathcal{I}. \tag{30}$$

In order to reach a contradiction to the first desired conclusion, suppose that $|\mathcal{S}^v| = \infty$. For any $k \in \mathcal{S}^v$, it follows from Lemmas 7 and 5(ii), and (14b) that

$$v_k^{\max} - v_{k+1}^{\max} \geq \kappa_\rho(1-\kappa_{v2})\|s_k\|^3 \geq \kappa_\rho(1-\kappa_{v2})\kappa_{ntn}^3\|n_k\|^3. \tag{31}$$

By Lemma 6, $0 < v_{k+1}^{\max} \leq v_k^{\max}$ for all $k \in \mathcal{I}$, meaning that $\{v_k^{\max} - v_{k+1}^{\max}\} \to 0$, which together with (31) shows that $\{\|n_k\|\}_{k\in\mathcal{S}^v} \to 0$, contradicting (30). This proves that $|\mathcal{S}^v| < \infty$. Now, in order to reach a contradiction to the second desired conclusion, suppose that $|\mathcal{S}^f| = \infty$. Since $|\mathcal{S}^v| < \infty$, we can assume without loss of generality that $\mathcal{S} = \mathcal{S}^f$. This means that the sequence $\{f_k\}$ is monotonically nonincreasing. Combining this with the fact that $\{f_k\}$ is bounded below under Assumptions 1 and 2, it follows that $\{f_k\} \to f_{low}$ for some $f_{low} \in (-\infty,\infty)$ and $\{f_k - f_{k+1}\} \to 0$. Using these facts, the inequality $\rho_k^f \geq \kappa_\rho$ for all $k \in \mathcal{S}^f$, and $|\mathcal{S}^f| = \infty$, it follows that $\{\kappa_\rho\|s_k\|^3\}_{k\in\mathcal{S}^f} \leq \{f_k - f_{k+1}\}_{k\in\mathcal{S}^f} \to 0$, which gives $\{\|s_k\|\}_{k\in\mathcal{S}^f} \to 0$. This, in turn, implies that $\{\|n_k\|\}_{k\in\mathcal{S}^f} \to 0$ because of Lemma 5(ii) and (14b), which contradicts (30). Hence, $|\mathcal{S}^f| < \infty$. ∎

We are now prepared to prove that Algorithm 1 terminates finitely.

THEOREM 21. *Algorithm* 1 *terminates finitely, i.e.,* $|\mathcal{I}| < \infty$.

Proof. Suppose by contradiction that $|\mathcal{I}| = \infty$. Let us consider two cases.

Case 1 ($|\mathcal{V}| = \infty$). Since $|\mathcal{S}| < \infty$, it follows that $|\mathcal{V} \setminus \mathcal{S}^v| = |\mathcal{C}^v \cup \mathcal{E}^v| = \infty$, which along with Lemma 13 implies that $|\mathcal{E}^v| < \infty$ while $|\mathcal{C}^v| = \infty$. It now follows from Lemma 17(i) that $\{\delta_k^v\} \to 0$, which contradicts Lemma 19.

Case 2 ($|\mathcal{V}| < \infty$). For this case, we may assume without loss of generality that $\mathcal{F} = \mathcal{I}$. This implies with Lemma 5(i) that $\delta_k^v = \delta_0^v$ and $n_k = n_0 \neq 0$ for all $k \in \mathcal{I}$.

It also implies from step 9 of Algorithm 1 that (15) holds for all $k \in \mathcal{I}$; in particular, from (15a) it means that $t_k \neq 0$ for all $k \in \mathcal{I}$. Now, from $|\mathcal{V}| < \infty$, $|\mathcal{S}| < \infty$, and Lemma 17(ii), it follows that $\{\delta_k^f\} \to 0$, which by (10) yields $\{\delta_k^s\} \to 0$. It then follows from step 20 of Algorithm 1 and $\mathcal{F} = \mathcal{I}$ that $\{n_k\} \to 0$, which contradicts our previous conclusion that $n_k = n_0 \neq 0$ for all $k \in \mathcal{I}$.

3.2. Complexity analysis for phase 1. Our goal in this subsection is to prove an upper bound on the total number of iterations required until phase 1 terminates, i.e., until the algorithm reaches $k \in \mathbb{N}$ such that $\|g_k^v\| \leq \epsilon$. To prove such a bound, we require the following additional assumption.

Assumption 22. The Hessian functions $H^v(x) := \nabla^2 v(x)$ and $\nabla^2 f(x)$ are Lipschitz continuous with constants $H_{Lip}^v \in (0, \infty)$ and $H_{Lip} \in (0, \infty)$, respectively, on a path defined by the sequence of iterates and trial steps computed in the algorithm.

Our first result in this subsection can be seen as a similar conclusion to that given by Lemma 16, but with this additional assumption in hand.

LEMMA 23. *For all $k \in \mathcal{I}$, if the trial step s_k and dual variable λ_k^v satisfy*

$$\lambda_k^v \geq \kappa_\delta^2 \kappa_{vm}^{-1} (H_{Lip}^v + 2\kappa_\rho) \|s_k\|, \tag{32}$$

then $\|n_k\| = \delta_k^v$ and $\rho_k^v \geq \kappa_\rho$.

Proof. For all $k \in \mathcal{I}$, there exists $\overline{x}_k$ on the line segment $[x_k, x_k + s_k]$ such that

$$m_k^v(s_k) - v(x_k + s_k) = \tfrac{1}{2} s_k^T \left(H_k^v - H^v(\overline{x}_k)\right) s_k \geq -\tfrac{1}{2} H_{Lip}^v \|s_k\|^3. \tag{33}$$

From this, (24b), and (6b), one deduces that

$$v(x_k) - v(x_k + s_k) \geq \tfrac{1}{2} \kappa_{vm} \lambda_k^v \|n_k\|^2 - \tfrac{1}{2} H_{Lip}^v \|s_k\|^3.$$

From Lemma 5(i), (32), and (6c), it follows that $\|n_k\| = \delta_k^v$, which along with (10) means that $\|s_k\| \leq \delta_k^s \leq \kappa_\delta \delta_k^v = \kappa_\delta \|n_k\|$, so, from above,

$$v(x_k) - v(x_k + s_k) \geq \tfrac{1}{2} \kappa_{vm} \kappa_\delta^{-2} \lambda_k^v \|s_k\|^2 - \tfrac{1}{2} H_{Lip}^v \|s_k\|^3. \tag{34}$$

From here, by steps 12 and 14 of Algorithm 1 and under (32), the result follows. $\square$

The next lemma reveals upper and lower bounds for an important ratio that will hold during the iteration immediately following a V-ITERATION contraction.

LEMMA 24. *For all $k \in \mathcal{C}^v$, it follows that*

$$\underline{\sigma} \leq \lambda_{k+1}^v / \|n_{k+1}\| \leq \max\left\{\overline{\sigma}, \gamma_\lambda \gamma_c^{-1} \lambda_k^v / \|n_k\|\right\}. \tag{35}$$

Proof. The result follows using the same logic as the proof of [17, Lemma 3.17]. $\square$

Now, we prove that the sequence $\{\sigma_k^v\}$ is bounded.

LEMMA 25. *There exists $\sigma_{\max}^v \in (0, \infty)$ such that $\sigma_k^v \leq \sigma_{\max}^v$ for all $k \in \mathcal{I}$.*

Proof. If $k \notin \mathcal{C}^v$, then steps 26 and 28 of Algorithm 1 give $\sigma_{k+1}^v \leftarrow \sigma_k^v$. Otherwise, if $k \in \mathcal{C}^v$, meaning that $\rho_k^v < \kappa_\rho$, then there are two cases to consider. If $k \in \mathcal{C}^v$ and $\lambda_k^v < \underline{\sigma}\|n_k\|$, then, by steps 23–27 of Algorithm 3, step 28 of Algorithm 1, and the fact that $\lambda_{k+1}^v = \lambda^v$ and $n_{k+1} = n(\lambda^v)$, where $(n(\lambda^v), \lambda^v)$ are computed either in steps 21, 22, or step 26 of Algorithm 3, it follows that $\sigma_{k+1}^v \leq \max\{\sigma_k^v, \overline{\sigma}\}$. Finally, if $k \in \mathcal{C}^v$ and $\lambda_k^v \geq \underline{\sigma}\|n_k\|$, then with Lemma 23 one has $\lambda_k^v < \kappa_\delta^2 \kappa_{vm}^{-1} (H_{Lip}^v + 2\kappa_\rho)\|s_k\|$. From the fact that $\lambda_k^v \geq \underline{\sigma}\|n_k\|$, Lemma 5(i), (6c), and (10), it follows that $\|s_k\| \leq \delta_k^s \leq \kappa_\delta \delta_k^v = \kappa_\delta \|n_k\|$. Hence, by step 28 of Algorithm 1 and Lemma 24, one finds that

$$\sigma_{k+1}^v \leftarrow \max\left\{\sigma_k^v, \lambda_{k+1}^v/\|n_{k+1}\|\right\} \leq \max\left\{\sigma_k^v, \overline{\sigma}, \gamma_\lambda \gamma_c^{-1} \kappa_\delta^3 \kappa_{vm}^{-1}(H_{Lip}^v + 2\kappa_\rho)\right\}.$$

Combining the results of these cases gives the desired conclusion. ∎

We now give a lower bound for the norm of some types of successful steps.

LEMMA 26. *For all $k \in \mathcal{S}_\sigma^v \cup \mathcal{S}^f$, the accepted step s_k satisfies*

$$\|s_k\| \geq (H_{Lip}^v + \kappa_{ht} + \sigma_{\max}^v/\kappa_{ntn}^2)^{-1/2}\|g_{k+1}^v\|^{1/2}. \tag{36}$$

Proof. Let $k \in \mathcal{S}_\sigma^v \cup \mathcal{S}^f$. It follows from (6a), the mean value theorem, the fact that $s_k = n_k + t_k$, Assumption 22, Lemma 5(ii), and (14c) that

$$\|g_{k+1}^v\| = \|g_{k+1}^v - g_k^v - (H_k^v + I\lambda_k^v)n_k\| \leq (H_{Lip}^v + \kappa_{ht})\|s_k\|^2 + \lambda_k^v\|n_k\|^2/\|n_k\|. \tag{37}$$

From step 2 of Algorithm 3 (if $k \in \mathcal{S}_\sigma^v$) and (15e) (if $k \in \mathcal{S}^f$), one finds that $\lambda_k^v/\|n_k\| \leq \sigma_k^v$. Combining this with (37), Lemma 5(ii), (14b), and Lemma 25, it follows that

$$\|g_{k+1}^v\| \leq H_{Lip}^v\|s_k\|^2 + \kappa_{ht}\|s_k\|^2 + \sigma_k^v\|n_k\|^2 \leq (H_{Lip}^v + \kappa_{ht} + \sigma_{\max}^v/\kappa_{ntn}^2)\|s_k\|^2,$$

which gives the desired result. ∎

We now give an iteration complexity result for a subset of successful iterations.

LEMMA 27. *For any $\epsilon \in (0,\infty)$, the total number of elements in*

$$\mathcal{K}(\epsilon) := \{k \in \mathcal{I} : k \geq 0 \text{ and } (k-1) \in \mathcal{S}_\sigma^v \cup \mathcal{S}^f\}$$

is at most

$$\left\lfloor \left(\frac{v_0^{\max}}{\kappa_\rho(1-\kappa_{v2})(H_{Lip}^v + \kappa_{ht} + \sigma_{\max}^v/\kappa_{ntn}^2)^{-3/2}} \right) \epsilon^{-3/2} \right\rfloor =: K_\sigma(\epsilon) \geq 0. \tag{38}$$

Proof. From Lemmas 7 and 26, it follows that, for all $k \in \mathcal{K}(\epsilon) \subseteq \mathcal{I}$,

$$\begin{aligned} v_{k-1}^{\max} - v_k^{\max} &\geq \kappa_\rho(1-\kappa_{v2})\|s_{k-1}\|^3 \\ &\geq \kappa_\rho(1-\kappa_{v2})(H_{Lip}^v + \kappa_{ht} + \sigma_{\max}^v/\kappa_{ntn}^2)^{-3/2}\epsilon^{3/2}. \end{aligned}$$

In addition, since $|\mathcal{K}(\epsilon)| < \infty$ follows by Theorem 21, the reduction in $v_k^{\max}$ obtained up to the largest index in $\mathcal{K}(\epsilon)$, call it $\bar{k}(\epsilon)$, satisfies

$$\begin{aligned} v_0^{\max} - v_{\bar{k}(\epsilon)}^{\max} &= \sum_{k=1}^{\bar{k}_\epsilon}(v_{k-1}^{\max} - v_k^{\max}) \geq \sum_{k\in\mathcal{K}(\epsilon)}(v_{k-1}^{\max} - v_k^{\max}) \\ &\geq |\mathcal{K}(\epsilon)|\kappa_\rho(1-\kappa_{v2})(H_{Lip}^v + \kappa_{ht} + \sigma_{\max}^v/\kappa_{ntn}^2)^{-3/2}\epsilon^{3/2}. \end{aligned}$$

Rearranging this inequality to yield an upper bound for $|\mathcal{K}(\epsilon)|$ and using the fact that $v_k^{\max} \geq 0$ for all $k \in \mathcal{I}$ (see Lemma 6), the desired result follows. ∎

In order to bound the total number of successful iterations in $\mathcal{I}$, we also need an upper bound for the cardinality of $\mathcal{S}_\Delta^v$. This is the subject of our next lemma.

LEMMA 28. *The cardinality of the set $\mathcal{S}_\Delta^v$ is bounded above by*

$$\left\lfloor \frac{v_0^{\max}}{\kappa_\rho \kappa_{ntn}^3 (1-\kappa_{v2})(\Delta_0^v)^3} \right\rfloor := K_\Delta^v \geq 0. \tag{39}$$

Proof. For all $k \in \mathcal{S}_\Delta^v \subseteq \mathcal{S}$, it follows from Lemmas 7 and 5(ii), (14b), and Lemma 10(ii) that the decrease in the trust funnel radius satisfies

$$\begin{aligned} v_k^{\max} - v_{k+1}^{\max} &\geq \kappa_\rho(1-\kappa_{v2})\|s_k\|^3 \geq \kappa_\rho \kappa_{ntn}^3 (1-\kappa_{v2})\|n_k\|^3 \\ &= \kappa_\rho \kappa_{ntn}^3 (1-\kappa_{v2})(\Delta_k^v)^3 \geq \kappa_\rho \kappa_{ntn}^3 (1-\kappa_{v2})(\Delta_0^v)^3. \end{aligned}$$

Now, using the fact that $\{v_k^{\max}\}$ is bounded below by zero (see Lemma 6), one finds that

$$v_0^{\max} \geq \sum_{k \in \mathcal{S}_\Delta^v} (v_k^{\max} - v_{k+1}^{\max}) \geq |\mathcal{S}_\Delta^v| \kappa_\rho \kappa_{ntn}^3 (1-\kappa_{v2})(\Delta_0^v)^3,$$

which gives the desired result. ◻

Having now provided upper bounds for the numbers of successful iterations, we need to bound the number of unsuccessful iterations in $\mathcal{I}$. To this end, first we prove that a critical ratio increases by at least a constant factor after an iteration in $\mathcal{C}^v$.

LEMMA 29. *If $k \in \mathcal{C}^v$ and $\lambda_k^v \geq \underline{\sigma}\|n_k\|$, then $\frac{\lambda_{k+1}^v}{\|n_{k+1}\|} \geq \min\{\gamma_\lambda, \frac{1}{\gamma_c}\} \frac{\lambda_k^v}{\|n_k\|}$.*

Proof. The proof follows the same logic as in [17, Lemma 3.23]. ◻

We are now able to provide an upper bound on the number of unsuccessful iterations in $\mathcal{C}^v$ that may occur between any two successful iterations.

LEMMA 30. *If $\bar{k} \in \mathcal{S} \cup \{0\}$, then*

$$|\mathcal{C}^v \cap \mathcal{I}_\mathcal{S}(\bar{k})| \leq 1 + \left\lfloor \frac{1}{\log(\min\{\gamma_\lambda, \gamma_c^{-1}\})} \log\left(\frac{\sigma_{\max}^v}{\underline{\sigma}}\right) \right\rfloor =: K_\mathcal{C}^v \geq 0. \tag{40}$$

Proof. The result holds trivially if $|\mathcal{C}^v \cap \mathcal{I}_\mathcal{S}(\bar{k})| = 0$. Thus, we may proceed under the assumption that $|\mathcal{C}^v \cap \mathcal{I}_\mathcal{S}(\bar{k})| \geq 1$. Let $k_{\mathcal{C}^v}$ be the smallest element in $\mathcal{C}^v \cap \mathcal{I}_\mathcal{S}(\bar{k})$. It then follows from Lemma 10(i) and (ii), Lemma 12, and step 12 of Algorithm 1 that for all $k \in \mathcal{I}$ satisfying $k_{\mathcal{C}^v} + 1 \leq k \leq k_\mathcal{S}(\bar{k})$ we have

$$\|n_k\| \leq \delta_k^v \leq \delta_{k_{\mathcal{C}^v}+1}^v < \delta_{k_{\mathcal{C}^v}}^v \leq \Delta_{k_{\mathcal{C}^v}}^v \leq \Delta_{k_\mathcal{S}(\bar{k})}^v,$$

which for $k = k_\mathcal{S}(\bar{k})$ means that $k_\mathcal{S}(\bar{k}) \in \mathcal{S}^f \cup \mathcal{S}_\sigma^v$. From Lemma 24, it follows that $\lambda_{k_{\mathcal{C}^v}+1}^v \geq \underline{\sigma}\|n_{k_{\mathcal{C}^v}+1}\|$, which by $k_\mathcal{S}(\bar{k}) \in \mathcal{S}^f \cup \mathcal{S}_\sigma^v$, Lemmas 25 and 29, (15e), step 2 of Algorithm 3, and the fact that $(n_{k+1}, \lambda_{k+1}^v) = (n_k, \lambda_k^v)$ for any $k \in \mathcal{C}^f$ means that

$$\sigma_{\max}^v \geq \sigma_{k_\mathcal{S}(\bar{k})}^v \geq \lambda_{k_\mathcal{S}(\bar{k})}^v / \|n_{k_\mathcal{S}(\bar{k})}\| \geq \left(\min\left\{\gamma_\lambda, \gamma_c^{-1}\right\}\right)^{|\mathcal{C}^v \cap \mathcal{I}_\mathcal{S}(\bar{k})|-1} \underline{\sigma},$$

from which the desired result follows. ◻

For our ultimate complexity result, the main component that remains to be proved is a bound on the number of unsuccessful iterations in $\mathcal{C}^f$ between any two successful iterations. To this end, we first need some preliminary results pertaining to the trial step and related quantities during an F-ITERATION. Our first such result pertains to the change in the objective function model yielded by the tangential step.

LEMMA 31. *For any $k \in \mathcal{I}$, the vectors n_k and t_k and dual variable λ_k^f satisfy*

$$m_k^f(n_k) - m_k^f(n_k + t_k) = \tfrac{1}{2} t_k^T (H_k + \lambda_k^f I) t_k + \tfrac{1}{2} \lambda_k^f \|t_k\|^2 + \lambda_k^f n_k^T t_k. \tag{41}$$

Proof. If $k \notin \mathcal{I}^t$ so that $t_k = 0$ and $\lambda_k^f = 0$ (by the COMPUTE_STEPS subroutine in Algorithm 1), then (41) trivially holds. Thus, for the remainder of the proof, let us assume that $k \in \mathcal{I}^t$. It now follows from the definition of m_k^f that

$$\begin{aligned} m_k^f(n_k) - m_k^f(n_k + t_k) &= -(g_k + H_k n_k)^T t_k - \tfrac{1}{2} t_k^T H_k t_k \\ &= -(g_k + (H_k + \lambda_k^f I) n_k + (H_k + \lambda_k^f I) t_k + J_k^T y_k^f)^T t_k \\ &\quad + \tfrac{1}{2} t_k^T (H_k + \lambda_k^f I) t_k + \tfrac{1}{2} \lambda_k^f \|t_k\|^2 + \lambda_k^f n_k^T t_k + (y_k^f)^T J_k t_k \\ &= \tfrac{1}{2} t_k^T (H_k + \lambda_k^f I) t_k + \tfrac{1}{2} \lambda_k^f \|t_k\|^2 + \lambda_k^f n_k^T t_k, \end{aligned}$$

where the last equality follows from (12a). □

The next lemma reveals that, for an F-ITERATION, if the dual variable for the tangential step trust region constraint is large enough, then the trust region constraint is active and the iteration will be successful.

LEMMA 32. *For all $k \in \mathcal{F}$, if the trial step s_k and the dual variable λ_k^f satisfy*

$$\lambda_k^f \geq (\kappa_{fm} \kappa_{st}^2 (1 - \kappa_{ntt}))^{-1} (\kappa_{hs} + H_{Lip} + 2\kappa_\rho) \|s_k\|, \tag{42}$$

then $\|s_k\| = \delta_k^s$ and $\rho_k^f \geq \kappa_\rho$.

Proof. Observe from (42) and Lemma 5(i) that $\lambda_k^f > 0$, which along with (12c) proves that $\|s_k\| = \delta_k^s$. Next, since $k \in \mathcal{F}$, it must mean that (15) is satisfied. It then follows from (15b), Lemma 31, (12), (15d), and (15a) that

$$\begin{aligned} m_k^f(0) - m_k^f(s_k) &\geq \kappa_{fm} (m_k^f(n_k) - m_k^f(s_k)) \\ &= \kappa_{fm} (\tfrac{1}{2} t_k^T (H_k + \lambda_k^f I) t_k + \tfrac{1}{2} \lambda_k^f \|t_k\|^2 + \lambda_k^f n_k^T t_k) \\ &\geq \kappa_{fm} (\tfrac{1}{2} - \tfrac{1}{2} \kappa_{ntt}) \lambda_k^f \|t_k\|^2 \geq \tfrac{1}{2} \kappa_{fm} \kappa_{st}^2 (1 - \kappa_{ntt}) \lambda_k^f \|s_k\|^2. \end{aligned} \tag{43}$$

Next, the mean value theorem gives the existence of an $\overline{x} \in [x_k, x_k + s_k]$ such that $f(x_k + s_k) = f_k + g_k^T s_k + \tfrac{1}{2} s_k^T \nabla^2 f(\overline{x}) s_k$, which with (15f) and Assumption 22 gives

$$\begin{aligned} & m_k^f(s_k) - f(x_k + s_k) \\ &= f_k + g_k^T s_k + \tfrac{1}{2} s_k^T H_k s_k - f_k - g_k^T s_k - \tfrac{1}{2} s_k^T \nabla^2 f(\overline{x}) s_k \\ &= \tfrac{1}{2} s_k^T \left(H_k - \nabla^2 f(x_k)\right) s_k + \tfrac{1}{2} s_k^T \left(\nabla^2 f(x_k) - \nabla^2 f(\overline{x})\right) s_k \\ &\geq -\tfrac{1}{2} \left\| \left(H_k - \nabla^2 f(x_k)\right) s_k \right\| \|s_k\| - \tfrac{1}{2} \left\| \left(\nabla^2 f(x_k) - \nabla^2 f(\overline{x})\right) s_k \right\| \|s_k\| \\ &\geq -\tfrac{1}{2} \left(\kappa_{hs} + H_{Lip}\right) \|s_k\|^3. \end{aligned}$$

Finally, combining the previous inequality, $f_k = m_k^f(0)$, and (43), one finds that

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - m_k^f(s_k) + m_k^f(s_k) - f(x_k + s_k) \\ &\geq \tfrac{1}{2} \kappa_{fm} \kappa_{st}^2 (1 - \kappa_{ntt}) \lambda_k^f \|s_k\|^2 - \tfrac{1}{2} (\kappa_{hs} + H_{Lip}) \|s_k\|^3, \end{aligned}$$

which combined with (42) shows that $\rho_k^f \geq \kappa_\rho$ as desired. □

We now show that a critical ratio increases by at least a constant factor after any unsuccessful F-ITERATION followed by an iteration in which a nonzero tangential step is computed and not reset to zero.

LEMMA 33. *If* $k \in \mathcal{C}^f$, $\lambda_k^f \geq \underline{\sigma}\|s_k\|$, *and* $(k+1) \in \mathcal{I}^t$, *then* $\frac{\lambda_{k+1}^f}{\|s_{k+1}\|} \geq \frac{\lambda_k^f}{\gamma_c\|s_k\|}$.

Proof. With Lemma 5(i), it follows that $\lambda_k^f \geq \underline{\sigma}\|s_k\| > 0$, meaning that $\|s_k\| = \delta_k^s$. In addition, since $k \in \mathcal{C}^f$, one finds that the condition in step 12 of Algorithm 2 tests false in iteration k. Hence, step 16 of Algorithm 2 is reached, meaning, with (10), that $\delta_{k+1}^f = \gamma_c\|s_k\| \leq \gamma_c \kappa_\delta \delta_k^v$. Then, from the facts that $\gamma_c < 1$ and $\delta_{k+1}^v \leftarrow \delta_k^v$ (see step 12 of Algorithm 1), it follows that $\delta_{k+1}^f \leq \kappa_\delta \delta_{k+1}^v$. Consequently, again with (10), it follows that $\|s_{k+1}\| = \delta_{k+1}^s = \delta_{k+1}^f = \gamma_c\|s_k\|$. Combining this with the fact that Lemma 11(i) yields $\lambda_{k+1}^f \geq \lambda_k^f$, the result follows. ☐

LEMMA 34. *If* $k \in \mathcal{C}^f$ *and* $(k+1) \in \mathcal{I}^t$, *then* $\underline{\sigma} \leq \lambda_{k+1}^f/\|s_{k+1}\|$.

Proof. Since $k \in \mathcal{C}^f$, there are two cases to consider.

Case 1 (step 14 of Algorithm 2 is reached). In this case, it follows that $\|s_{k+1}\| = \delta_{k+1}^f = \|n_k + t(\lambda^f)\|$ with $(t(\lambda^f), \lambda^f)$ computed in step 13 of Algorithm 2. Together with the fact that $(k+1) \in \mathcal{I}^t$, it follows that $\lambda_{k+1}^f/\|s_{k+1}\| = \lambda^f/\|n_k + t(\lambda^f)\| \geq \underline{\sigma}$.

Case 2 (step 14 of Algorithm 2 is not reached). This only happens if the condition in step 12 of Algorithm 2 tested false, meaning that $\lambda_k^f/\|s_k\| \geq \underline{\sigma}$. Hence, from Lemma 33, it follows that $\lambda_{k+1}^f/\|s_{k+1}\| \geq \lambda_k^f/\gamma_c\|s_k\|$, which by the facts that $\gamma_c < 1$ and $\lambda_k^f/\|s_k\| \geq \underline{\sigma}$ gives the desired result.

Next, we provide a bound on the number of iterations in $\mathcal{C}^f$ that may occur before the first or between consecutive iterations in the set $\mathcal{S} \cup \mathcal{V}$.

LEMMA 35. *If* $\bar{k} \in \mathcal{S} \cup \mathcal{V} \cup \{0\}$, *then*

$$|\mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k})| \leq 2 + \left\lfloor \frac{1}{\log(\gamma_c^{-1})} \log\left(\frac{\kappa_{hs} + H_{Lip} + 2\kappa_\rho}{\underline{\sigma}\kappa_{fm}\kappa_{st}^2(1-\kappa_{ntt})}\right)\right\rfloor =: K_{\mathcal{C}}^f \geq 0. \tag{44}$$

Proof. Let $\bar{k} \in \mathcal{S} \cup \mathcal{V} \cup \{0\}$. Then, $\mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k}) \subseteq \mathcal{C}^f$. The result follows trivially if $|\mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k})| \leq 1$. Therefore, for the remainder of the proof, let us assume that $|\mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k})| \geq 2$. It follows from Lemma 34, $\bar{k}+1 \in \mathcal{C}^f$, and $\bar{k}+2 \in \mathcal{C}^f \subseteq \mathcal{F}$ (meaning that $t_{\bar{k}+2} \neq 0$ and $(\bar{k}+2) \in \mathcal{I}^t$) that $\underline{\sigma} \leq \lambda_{\bar{k}+2}^f/\|s_{\bar{k}+2}\|$. Combining this inequality with Lemmas 32 and 33, and the fact that $(k_{\mathcal{S}\cup\mathcal{V}}(\bar{k})-1) \in \mathcal{C}^f$, yields

$$\underline{\sigma}\left(\frac{1}{\gamma_c}\right)^{(k_{\mathcal{S}\cup\mathcal{V}}(\bar{k})-1)-(\bar{k}+2)} \leq \frac{\lambda_{k_{\mathcal{S}\cup\mathcal{V}}(\bar{k})-1}^f}{\|s_{k_{\mathcal{S}\cup\mathcal{V}}(\bar{k})-1}\|} \leq \left(\frac{\kappa_{hs} + H_{Lip} + 2\kappa_\rho}{\kappa_{fm}\kappa_{st}^2(1-\kappa_{ntt})}\right).$$

The desired result now follows since $|\mathcal{I}_{\mathcal{S}\cup\mathcal{V}}(\bar{k})| = k_{\mathcal{S}\cup\mathcal{V}}(\bar{k}) - \bar{k} - 1$. ☐

We have now arrived at our complexity result for phase 1.

THEOREM 36. *For a scalar* $\epsilon \in (0,\infty)$, *the cardinality of* $\mathcal{I}$ *is at most*

$$K(\epsilon) := 1 + (K_\sigma(\epsilon) + K_\Delta^v)(K_{\mathcal{C}}^v + 1)K_{\mathcal{C}}^f, \tag{45}$$

where $K_\sigma(\epsilon)$, K_Δ^v, $K_{\mathcal{C}}^v$, *and* $K_{\mathcal{C}}^f$ *are defined in Lemmas* 27, 28, 30, *and* 35, *respectively. Consequently, for any* $\bar{\epsilon} \in (0,\infty)$, *it follows that* $K(\epsilon) = \mathcal{O}(\epsilon^{-3/2})$ *for all* $\epsilon \in (0,\bar{\epsilon})$.

Proof. Without loss of generality, let us assume that at least one iteration is performed. Then, Lemmas 27 and 28 guarantee that at most $K_\sigma(\epsilon) + K_\Delta^v$ successful iterations are included in $\mathcal{I}$. In addition, Lemmas 13, 30, and 35 guarantee that, before each successful iteration, there can be at most $(K_\mathcal{C}^v + 1)K_\mathcal{C}^f$ unsuccessful iterations. Also accounting for the first iteration, the desired result follows. ◻

Building on this theorem, one may now apply the analysis in [10, sections 3.1–3.2] to arrive at the following corollary involving practical termination conditions for phase 1.

COROLLARY 37. *Let $(\epsilon_{feas}, \epsilon_{inf}) \in (0,1) \times (0,1)$ be given constants. Then, at most $\mathcal{O}(\max\{\epsilon_{feas}^{-1/2}, \epsilon_{inf}^{-3/2}\})$ iterations are required until either*

$$\|c_k\| \leq \epsilon_{feas} \tag{46a}$$

$$\text{or} \quad \|J_k^T c_k\| \leq \epsilon_{inf}\|c_k\|. \tag{46b}$$

Proof. The result follows in the same manner as [10, Theorem 3.2]. In particular, the only property of the phase 1 algorithm needed for the proof in [10] is that reductions in v for successful steps are at least a fraction of $\|g_k^v\|^{3/2}$. The same holds for our algorithm with respect to $\{v_k^{\max}\}$, as shown by Lemmas 7 and 26. ◻

If the constraint Jacobians encountered by the algorithm are not rank deficient (and do not tend toward rank deficiency), then the following corollary gives a similar result to that above, but for an infeasibility measure. This occurs, e.g., if all iterates and all limit points of the algorithm are points at which the linear independence constraint qualification (LICQ) is satisfied.

COROLLARY 38. *Suppose that, for all $k \in \mathbb{N}$, the constraint Jacobian J_k has full row rank with singular values bounded below by $\zeta_{\min} \in (0,\infty)$. Then, for $\epsilon \in (0,\infty)$, the cardinality of $\mathcal{I}_c := \{k \in \mathbb{N} : \|c_k\| > \epsilon/\zeta_{\min}\}$ is at most $K(\epsilon)$, defined in (45). Consequently, for any $\bar{\epsilon} \in (0,\infty)$, the cardinality of $\mathcal{I}_c$ is $\mathcal{O}(\epsilon^{-3/2})$ for all $\epsilon \in (0,\bar{\epsilon})$.*

Proof. Under the stated conditions, $\|g_k^v\| \equiv \|J_k^T c_k\| \geq \zeta_{\min}\|c_k\|$ for all $k \in \mathcal{I}$. Thus, since $\|g_k^v\| \leq \epsilon$ implies $\|c_k\| \leq \epsilon/\zeta_{\min}$, the result follows from Theorem 36. ◻

4. Phase 2: Obtaining optimality.

A complete algorithm for solving problem (1) proceeds as follows. The phase 1 method, Algorithm 1, is run until either an approximate feasible point or approximate infeasible stationary point is found, i.e., for some $(\epsilon_{feas}, \epsilon_{inf}) \in (0,1) \times (0,1)$, the method is run until (46) holds for some $k \in \mathbb{N}$. If phase 1 terminates with (46a) failing to hold and (46b) holding, then the entire algorithm is terminated with a declaration of having found an infeasible (approximately) stationary point. Otherwise, if (46a) holds, then a phase 2 method is run that maintains at least ϵ_{feas}-feasibility while seeking optimality.

There are various options for phase 2. For example, respecting the current state-of-the-art nonlinear optimization methods, one can run a trust funnel method such as that in [23]. One can even run such a method with the initial trust funnel radius for $v(x) = \frac{1}{2}\|c(x)\|^2$ set at $\frac{1}{2}\epsilon_{feas}^2$ so that ϵ_{feas}-feasibility will be maintained as optimality is sought. We do not claim worst-case iteration complexity guarantees for such a method, though empirical evidence suggests that it would perform well; see section 6. If c is affine, then one could run a method, such as the ARC method from [5, 6] (see also the previous work in [24, 31, 35]) or the TRACE method from [17], where steps toward reducing the objective are restricted to the null space of the constraint Jacobian. For such a reduced-space method, ϵ_{feas}-feasibility will be maintained while the analyses in [5, 6, 17] guarantee that the number of iterations required to reduce the norm of the reduced gradient below a given tolerance $\epsilon_{opt} \in (0, \infty)$ is at most $\mathcal{O}(\epsilon_{opt}^{-3/2})$. With $\epsilon = \epsilon_{opt} = \epsilon_{feas}$, this gives an overall (phase 1 + phase 2) complexity of $\mathcal{O}(\epsilon^{-3/2})$.

More interesting for our purposes are phase 2 approaches designed with an eye toward attaining good complexity properties. To achieve this, one can run the objective-target-following approach stated as [10, Algorithm 4.1, phase 2] or as [2, Algorithm 2.1, phase 2]. These approaches apply an unconstrained optimization algorithm to minimize the residual $\Phi : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}$ defined by $\Phi(x,t) = \frac{1}{2}\|r(x,t)\|^2$, where $r : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}$ is defined by

$$r(x,t) = (c(x), f(x) - t). \tag{47}$$

Updated dynamically by the algorithm, the parameter t may be viewed as a target value for reducing the objective function value.

Our goal here is to show that a phase 2 method can be built upon the TRACE method from [17] yielding the same worst-case complexity properties as the ARC-based method in [10]. For our algorithm and analysis of it, we assume the following.

Assumption 39. The initial objective value for phase 2, namely, f_0, is bounded above independently from ϵ_{feas}. For all $x \in \mathbb{R}^N$ with $\|c(x)\| \leq \epsilon_{feas} \in (0, \infty)$, the objective f is bounded below by $f_{\min} \in \mathbb{R}$. The functions f and c and their first and second derivatives are Lipschitz continuous on the path defined by all phase 2 iterates.

Our phase 2 algorithm is stated as Algorithm 4. We refer the reader to [10] for further details on the design of the algorithm and to [17] for further details on TRACE. In short, in iteration $k \in \mathbb{N}$, the subsequent iterate x_{k+1} is computed to reduce $\Phi(\cdot, t_k)$ while the subsequent target t_{k+1} is chosen to ensure the relationships in the following lemma (whose proof follows that of [10, Lemma 4.1]).

Algorithm 4 TRACE algorithm for phase 2

Require: termination tolerance $\epsilon \in (0, \infty)$ and $x_0 \in \mathbb{R}^N$ with $\|c_0\| \leq \epsilon_{feas} \in (0, 1)$

procedure TRACE_PHASE_2
  set $t_0 \leftarrow f_0 - \sqrt{\epsilon_{feas}^2 - \|c_0\|^2}$
  for $k \in \mathbb{N}$ **do**
    perform one iteration of TRACE toward minimizing $\Phi(x, t_k)$ to compute s_k
    if s_k is an acceptable step **then**
      set $x_{k+1} \leftarrow x_k + s_k$ (and other quantities following TRACE)
      if $r(x_{k+1}, t_k) \neq 0$ and $\|\nabla_x \Phi(x_{k+1}, t_k)\| \leq \epsilon \|r(x_{k+1}, t_k)\|$ **then**
        terminate
      else
        set $t_{k+1} \leftarrow f(x_{k+1}) - \sqrt{\|r(x_k, t_k)\|^2 - \|r(x_{k+1}, t_k)\|^2 + (f(x_{k+1}) - t_k)^2}$
    else
      set $x_{k+1} \leftarrow x_k$ (and other quantities following TRACE)
      set $t_{k+1} \leftarrow t_k$

LEMMA 40. *Let* $\epsilon_{feas} \in (0,1)$ *be given. For all* $k \in \mathbb{N}$, *it follows that*

$$t_{k+1} \le t_k, \tag{48a}$$

$$0 \le f(x_k) - t_k \le \epsilon_{feas}, \tag{48b}$$

$$\|r(x_k, t_k)\| = \epsilon_{feas}, \tag{48c}$$

$$\text{and} \quad \|c(x_k)\| \le \epsilon_{feas}. \tag{48d}$$

Proof. Note that, in TRACE, the objective function is monotonically nonincreasing; see [17, equation (2.5); Algorithm 1, step 5]. Hence, each acceptable step s_k computed in Algorithm 4 yields $\Phi(x_{k+1}, t_k) \le \Phi(x_k, t_k)$, from which it follows that the value for t_{k+1} in step 10 is well defined. Then, since all definitions and procedures in Algorithm 4 that yield (48) are exactly the same as in [10, Algorithm 4.1], a proof for the inequalities in (48) is given by the proof of [10, Lemma 4.1]. ◻

In the next lemma, we recall a critical result from [17], arguing that it remains true for Algorithm 4 under our assumptions about the problem functions.

LEMMA 41. *Let* $\{\sigma_k\}$ *be generated as in* TRACE [17]. *Then, there exists a scalar constant* $\sigma_{\max} \in (0, \infty)$ *such that* $\sigma_k \le \sigma_{\max}$ *for all* $k \in \mathbb{N}$.

Proof. The result follows similarly to [17, Lemma 3.18]. Here, similar to [10, section 5], it is important to note that Assumption 39 ensures that Φ and its first and second derivatives are globally Lipschitz continuous on a path defined by the phase 2 iterates. This ensures that results of the kind given as [17, Lemmas 3.16 and 3.17] hold true, which are necessary for proving [17, Lemma 3.18] (for bounding $\{\sigma_k\}$ above by a $\sigma_{\max} \in (0, \infty)$). ◻

We now argue that the number of iterations taken for any fixed value of the target for the objective function is bounded above by a positive constant.

LEMMA 42. *The number of iterations required before the first accepted step or between two successive accepted steps with a fixed target is bounded above by*

$$K_t := 2 + \left\lfloor \frac{1}{\log(\min\{\gamma_\lambda, \gamma_c^{-1}\})} \log\left(\frac{\sigma_{\max}}{\underline{\sigma}}\right) \right\rfloor,$$

where the constants $\gamma_\lambda \in (0, \infty)$, $\gamma_c \in (0,1)$, *and* $\underline{\sigma} \in (0, \infty)$ *are parameters used by* TRACE *(see* [17, Algorithm 1]*) that are independent of* k *and satisfy* $\underline{\sigma} \le \sigma_{\max}$.

Proof. The properties of TRACE corresponding to so-called *contraction* and *expansion* iterations all hold for Algorithm 4 for sequences of iterations in which a target value is held fixed. Therefore, the result follows by [17, Lemmas 3.22 and 3.24], which combined show that the maximum number of iterations of interest is equal to the maximum number of contractions that may occur plus one. ◻

The next lemma merely states a fundamental property of TRACE.

LEMMA 43. *Let* $H_\Phi \in (0, \infty)$ *be the Lipschitz constant for the Hessian function of* Φ *along the path of phase* 2 *iterates and let* $\eta \in (0,1)$ *be the acceptance constant from* TRACE. *Then, for* x_{k+1} *following an accepted step* s_k, *it follows that*

$$\Phi(x_k, t_k) - \Phi(x_{k+1}, t_k) \ge \eta(H_\Phi + \sigma_{\max})^{-3/2} \|\nabla_x \Phi(x_{k+1}, t_k)\|^{3/2}.$$

Proof. Using Lemma 41 and adapting the conclusion of [17, Lemma 3.19], the result follows as in the beginning of the proof of [17, Lemma 3.20]. ◻

The preceding lemma allows us to prove the following useful result.

LEMMA 44. *For x_{k+1} following an accepted step s_k yielding*

$$\|\nabla_x \Phi(x_{k+1}, t_k)\| > \epsilon \|r(x_{k+1}, t_k)\|, \tag{49}$$

it follows that $\|r(x_k, t_k)\| - \|r(x_{k+1}, t_k)\| \geq \kappa_t \min\{\epsilon^{3/2}\epsilon_{feas}^{1/2}, \epsilon_{feas}\}$, where $\beta \in (0,1)$ is any fixed problem-independent constant, $\omega := \eta(H_\Phi + \sigma_{\max})^{-3/2} \in (0, \infty)$ is the constant appearing in Lemma 43, and $\kappa_t := \min\{\omega\beta^{3/2}, 1 - \beta\}$.

Proof. The proof follows similarly to that of [10, Lemma 5.3] using the result of Lemma 43 and (48c); for a complete proof, see [18]. ◻

We next prove a lower bound on the decrease of the target value.

LEMMA 45. *Suppose that the termination tolerance is set so that $\epsilon \leq \epsilon_{feas}^{1/3}$. Then, for x_{k+1} following an accepted step s_k such that the termination conditions in step 7 are not satisfied, it follows that, with $\kappa_t \in (0,1)$ defined as in Lemma 44,*

$$t_k - t_{k+1} \geq \kappa_t \epsilon^{3/2} \epsilon_{feas}^{1/2}. \tag{50}$$

Proof. The proof follows similarly to that of [10, Lemma 5.3]. In particular, if the reason that the termination conditions in step 7 are not satisfied is because (49) holds, then Lemma 44 and the fact that $\epsilon \leq \epsilon_{feas}^{1/3}$ imply that

$$\|r(x_k, t_k)\| - \|r(x_{k+1}, t_k)\| \geq \kappa_t \min\{\epsilon^{3/2}\epsilon_{feas}^{1/2}, \epsilon_{feas}\} \geq \kappa_t \epsilon^{3/2}\epsilon_{feas}^{1/2}.$$

On the other hand, if the condition in step 7 is not satisfied because $\|r(x_{k+1}, t_k)\| = 0$, it follows from (48c), $\kappa_t \in (0,1)$, and $\epsilon \leq \epsilon_{feas}^{1/3}$ that $\|r(x_k, t_k)\| - \|r(x_{k+1}, t_k)\| = \epsilon_{feas} \geq \kappa_t \epsilon^{3/2}\epsilon_{feas}^{1/2}$. Combining these two cases, (47), and (48c), one finds that

$$(f(x_{k+1}) - t_k)^2 + \|c(x_{k+1})\|^2 = \|r(x_{k+1}, t_k)\|^2 \leq (\epsilon_{feas} - \kappa_t \epsilon^{3/2}\epsilon_{feas}^{1/2})^2.$$

Now, from step 10 of Algorithm 4, (47), and (48c), it follows that

$$\begin{aligned} t_k - t_{k+1} &= -(f(x_{k+1}) - t_k) + \sqrt{\|r(x_k, t_k)\|^2 - \|r(x_{k+1}, t_k)\|^2 + (f(x_{k+1}) - t_k)^2} \\ &= -(f(x_{k+1}) - t_k) + \sqrt{\|r(x_k, t_k)\|^2 - \|c(x_{k+1})\|^2} \\ &= -(f(x_{k+1}) - t_k) + \sqrt{\epsilon_{feas}^2 - \|c(x_{k+1})\|^2}. \end{aligned}$$

Overall, it follows that [10, Lemma 5.2] can be applied (with "f" $= f(x_{k+1}) - t_k$, "c" $= \|c(x_{k+1})\|$, "ϵ" $= \epsilon_{feas}$, and "τ" $= \epsilon_{feas} - \kappa_t \epsilon^{3/2}\epsilon_{feas}^{1/2}$) to obtain the result. ◻

We now show that if the termination condition in step 7 of Algorithm 4 is never satisfied, then the algorithm takes infinitely many accepted steps.

LEMMA 46. *If Algorithm 4 does not terminate finitely, then it takes infinitely many accepted steps.*

Proof. To derive a contradiction, suppose that the number of accepted steps is finite. Then, since it does not terminate finitely, there exists $\bar{k} \in \mathbb{N}$ such that s_k is not acceptable for all $k \geq \bar{k}$. Therefore, by the construction of the algorithm, it follows that $t_k = t_{\bar{k}}$ for all $k \geq \bar{k}$. This means that the algorithm proceeds as if the TRACE algorithm from [17] is being employed to minimize $\Phi(\cdot, t_{\bar{k}})$, from which it follows by [17, Lemmas 3.7, 3.8, and 3.9] that, for some sufficiently large $k \geq \bar{k}$, an acceptable step s_k will be computed. This contradiction completes the proof. ◻

Before proceeding, let us discuss the situation in which the termination conditions in step 7 of Algorithm 4 are satisfied. This discussion, originally presented in [10], justifies the use of these termination conditions.

Suppose $\|r(x_{k+1}, t_k)\| \neq 0$ and $\|\nabla_x \Phi(x_{k+1}, t_k)\| \leq \epsilon \|r(x_{k+1}, t_k)\|$. If $f_{k+1} = t_k$, then these mean that $\|c_{k+1}\| \neq 0$ and $\|\nabla_x \Phi(x_{k+1}, t_k)\| \leq \epsilon \|c_{k+1}\|$, which along with $\nabla_x \Phi(x_{k+1}, t_k) = J_{k+1}^T c_{k+1} + (f_{k+1} - t_k) g_{k+1}$ would imply that $\|J_{k+1}^T c_{k+1}\| \leq \epsilon \|c_{k+1}\|$. That is, under these conditions, x_{k+1} is an approximate first-order stationary point for minimizing $\|c\|$. If $f_{k+1} \neq t_k$, then the satisfied termination conditions imply that $\|J_{k+1}^T c_{k+1} + (f_{k+1} - t_k) g_{k+1}\| / \|r(x_{k+1}, t_k)\| \leq \epsilon$. By dividing the numerator and denominator of the left-hand side by $f(x_{k+1}) - t_k > 0$ (recall (48b)), defining

$$(51) \qquad y(x_{k+1}, t_k) := c(x_{k+1}) / (f(x_{k+1}) - t_k) \in \mathbb{R}^M,$$

and substituting $y(x_{k+1}, t_k)$ back into the inequality, one finds that

$$(52) \qquad \|J_{k+1}^T y(x_{k+1}, t_k) + g_{k+1}\| / \|(y(x_{k+1}, t_k), 1)\| \leq \epsilon.$$

As argued in [10], we may use a perturbation argument to say that if $(x_{k+1}, y(x_{k+1}, t_k))$ satisfies the *relative KKT error* condition (52) and $\|c_{k+1}\| \leq \epsilon_{feas}$, then it corresponds to a first-order stationary point for problem (1). Specifically, consider $x = x_* + \delta_x$ and $y = y_* + \delta_y$, where (x_*, y_*) is a primal-dual pair satisfying the KKT conditions for problem (1). Then, a first-order Taylor expansion of $J(x_*)^T y_* + g(x_*)$ to estimate its value at (x, y) yields the estimate $(\nabla^2 f(x_*) + \sum_{i=1}^M [y_i]_* \nabla^2 c_i(x^*)) \delta_x + J(x^*)^T \delta_y$. The presence of the dual variable y^* in this estimate confirms that the magnitude of the dual variable should not be ignored in a relative KKT error condition such as (52).

We now prove our worst-case iteration complexity result for phase 2.

THEOREM 47. *Suppose that the termination tolerances are set so that* $\epsilon \leq \epsilon_{feas}^{1/3}$ *with* $\epsilon_{feas} \in (0, 1)$. *Then, Algorithm* 4 *requires at most* $\mathcal{O}(\epsilon^{-3/2} \epsilon_{feas}^{-1/2})$ *iterations until the termination condition in step* 7 *is satisfied, at which point either*

$$(53a) \qquad \|J_{k+1}^T y(x_{k+1}, t_k) + g_{k+1}\| / \|(y(x_{k+1}, t_k), 1)\| \leq \epsilon \text{ and } \|c_{k+1}\| \leq \epsilon_{feas}$$

$$(53b) \qquad \text{or } \|J_{k+1}^T c_{k+1}\| / \|c_{k+1}\| \leq \epsilon \text{ and } \|c_{k+1}\| \leq \epsilon_{feas}$$

is satisfied, with $y(x_{k+1}, t_k)$ *defined in* (51).

Proof. Recall that if the termination condition in step 7 is satisfied for some $k \in \mathbb{N}$, then, by the arguments prior to the lemma, either (53a) or (53b) will be satisfied. Thus, we aim to show an upper bound on the number of iterations required by the algorithm until the termination condition in step 7 is satisfied.

Without loss of generality, let us suppose that the algorithm performs at least one iteration. Then, we claim that there exists some $k \in \mathbb{N}$ such that the termination condition does not hold for (x_k, t_{k-1}), but does hold for (x_{k+1}, t_k). To see this, suppose for contradiction that the termination condition is never satisfied. Then, by Lemma 45, it follows that for all $k \in \mathbb{N}$ such that s_k is acceptable one finds that (50) holds. This, along with Lemma 46, implies that $\{t_k\} \searrow -\infty$. However, this along with (48b) implies that $\{f_k\} \searrow -\infty$, which contradicts Assumption 39.

Now, since the termination condition is satisfied at (x_{k+1}, t_k), but not in the iteration before, it follows that s_k must be an acceptable step. Hence, from Assumption 39, (48b), Lemma 45, and Algorithm 4, step 2, it follows that

$$f_{\min} \leq f_k \leq t_k + \epsilon_{feas} \leq t_0 - K_{\mathcal{A}} \kappa_t \epsilon^{3/2} \epsilon_{feas}^{1/2} + \epsilon_{feas} \leq f(x_0) - K_{\mathcal{A}} \kappa_t \epsilon^{3/2} \epsilon_{feas}^{1/2} + \epsilon_{feas},$$

where $K_{\mathcal{A}}$ is the number of accepted steps prior to iteration $(k+1)$. Rearranging, and since $\epsilon_{feas} \in (0,1)$, one finds that $K_{\mathcal{A}} \le \lceil (f_0 + \epsilon_{feas} - f_{\min} + 1)/(\kappa_t \epsilon^{3/2} \epsilon_{feas}^{1/2}) \rceil$. From this and Lemma 42, the desired result follows. □

This should be viewed in two ways. First, if $\epsilon = \epsilon_{feas}^{2/3}$, then the overall complexity is $\mathcal{O}(\epsilon_{feas}^{-3/2})$, though of course this corresponds to a looser tolerance on the relative KKT error than on feasibility. Second, if $\epsilon = \epsilon_{feas}$ (so that the two tolerances are equal), then the overall complexity is $\mathcal{O}(\epsilon_{feas}^{-2})$.

5. Numerical experiments. Our goal now is to demonstrate that instead of having a phase 1 method that solely seeks (approximate) feasibility (such as in [2, 10]), it is beneficial to employ a phase 1 method that also attempts to reduce the objective. To show this, a MATLAB implementation of our phase 1 (Algorithm 1) plus a phase 2 has been written. The implementation has two modes: one following Algorithm 1 and one employing the same procedures except that the tangential step t_k is set to zero for all $k \in \mathbb{N}$ so that all iterations are V-ITERATIONS. We refer to the former implementation as TF and the latter as TF-V-ONLY. For phase 2 for both methods, we implemented a trust funnel method based on that proposed in [23] with the modification that the normal step computation is never skipped. The initial value of $v_k^{\max}$ for phase 2 is the final value obtained from phase 1; note that this means that phase 2 does not necessarily maintain (near) feasibility as would Algorithm 4. In both phases 1 and 2, subproblems are solved to high accuracy using a MATLAB implementation of the trust region subproblem solver described in [12, Algorithm 7.3.4], which goes back to the work in [30]. However, if a step is rejected and a contraction is performed, then, when possible, rather than solve a trust region subproblem, we compute steps by solving linear systems (to solve $\mathcal{Q}_k^v(\cdot)$ and/or $\mathcal{Q}_k^f(\cdot)$ for a given dual variable). The fact that the normal step computation is never skipped and the subproblems are always solved to high accuracy allows our implementation to ignore so-called "y-iterations" [23].

Phase 1 in each implementation terminates in iteration $k \in \mathbb{N}$ if either

$$\|c_k\|_\infty \le 10^{-6} \max\{\|c_0\|_\infty, 1\} \quad \text{or} \quad \begin{cases} \|J_k^T c_k\|_\infty \le 10^{-9} \max\{\|J_0^T c_0\|_\infty, 1\} \\ \text{and } \|c_k\|_\infty > 10^{-3} \max\{\|c_0\|_\infty, 1\}. \end{cases} \tag{54}$$

Phase 2 terminates in iteration $k \in \mathbb{N}$ if the second condition in (54) holds or if both the first condition in (54) holds and, with y_k computed as least squares multipliers for all $k \in \mathbb{N}$, one finds $\|g_k + J_k^T y_k\|_\infty \le 10^{-9} \max\{\|g_0 + J_0^T y_0\|_\infty, 1\}$. Input parameters used in the code are stated in Table 1. The only values that do not appear are κ_ρ and γ_c. For κ_ρ, for simplicity we employed this in both (15c) and (17), as well as in the step acceptance conditions in step 2 of Algorithm 2 and step 2 of Algorithm 3. However, it turns out that our analysis can be adapted to handle different values in these places. Along these lines, our code uses $\kappa_\rho = 10^{-12}$ in (15c) and (17), but uses $\kappa_\rho = 10^{-8}$ in the step acceptance conditions. For γ_c, our code uses 0.5 in Algorithm 2 and 10^{-2} in Algorithm 3, where again our analysis can be extended to allow the use of different constants in these places. We chose these constant values since they worked well for both algorithms in our tests. For all $k \in \mathbb{N}$, we set H_k as the Hessian of the Lagrangian, which is set using least squares multipliers.

We ran TF and TF-V-ONLY to solve the equality constrained problems in the CUTEst test set [19]. In particular, we ran our experiments on a machine with 8GB of memory and set a time limit of four hours. Among 190 such problems, we removed 78 that had a constant (or null) objective, 17 for which phase 1 of both algorithms

TABLE 1
Input parameters for TF *and* TF-V-ONLY.

κ_n	9e-01	κ_{st}	1e-12	κ_p	1e-06	κ_δ	1e+02
κ_{vm}	1e-12	κ_{ntt}	1-(2e-12)	κ_{ht}	1e+20	γ_e	2e+00
κ_{ntn}	1e-12	κ_{v1}	9e-01	κ_{hs}	1e+20	γ_λ	2e+00
κ_{fm}	1e-12	κ_{v2}	9e-01	$\underline{\sigma}$	1e-12	$\overline{\sigma}$	1e+20

terminated immediately at the initial point due to the first condition in (54), one for which phase 1 of both algorithms terminated immediately at the initial point due to the second condition in (54), 26 for which both algorithms had insufficient memory, 19 on which both algorithms exceeded our time limit, and 8 problems for which both algorithms failed (due to different combinations of the reasons above, namely, memory limit, time limit, etc.). In addition, one problem was removed because TF found a relative stationary point but TF-V-ONLY failed due to a subproblem solver error, and two were removed because TF-V-ONLY found relative stationary points but TF failed because of our memory limit. The remaining set consisted of 38 problems.

TABLE 2
Numerical results for TF *and* TF-V-ONLY.

			TF						TF-V-ONLY				
			Phase 1				Phase 2		Phase 1			Phase 2	
Problem	n	m	#V	#F	f	$\|g + J^Ty\|$	#V	#F	#V	f	$\|g + J^Ty\|$	#V	#F
BT1	2	1	4	0	-8.02e-01	+4.79e-01	0	139	4	-8.00e-01	+7.04e-01	7	137
BT10	2	2	10	0	-1.00e+00	+5.39e-04	1	0	10	-1.00e+00	+6.74e-05	1	0
BT11	5	3	6	1	+8.25e-01	+4.84e-03	2	0	1	+4.55e+04	+2.57e+04	16	36
BT12	5	3	46	6	+6.57e+00	+2.06e-01	6	11	16	+3.34e+01	+4.15e+00	4	8
BT2	3	1	22	8	+1.45e+03	+3.30e+02	3	12	20	+6.14e+04	+1.82e+04	0	40
BT3	5	3	1	0	+4.09e+00	+6.43e+02	1	0	1	+1.01e+05	+8.89e+02	0	1
BT4	3	2	1	0	-1.86e+01	+9.77e+00	20	11	1	-1.86e+01	+9.77e+00	20	11
BT5	3	2	13	6	+9.67e+02	+4.75e+00	4	74	8	+9.62e+02	+7.38e-01	5	1
BT6	5	2	11	45	+2.77e-01	+4.64e-02	1	0	14	+5.81e+02	+4.50e+02	5	59
BT7	5	3	15	6	+1.31e+01	+5.57e+00	5	1	12	+1.81e+01	+1.02e+01	19	28
BT8	5	2	22	2	+1.00e+00	+4.29e-04	0	1	10	+2.00e+00	+2.00e+00	1	97
BT9	4	2	11	1	-1.00e+00	+8.56e-05	1	0	10	-9.69e-01	+2.26e-01	5	1
BYRDSPHR	3	2	29	2	-4.68e+00	+1.28e-05	0	0	19	-5.00e-01	+1.00e+00	16	5
CHAIN	800	401	9	0	+5.12e+00	+2.35e-04	3	20	9	+5.12e+00	+2.35e-04	3	20
FLT	2	2	15	4	+2.68e+10	+3.28e+05	0	13	19	+2.68e+10	+3.28e+05	0	17
GENHS28	10	8	1	0	+9.27e-01	+5.83e+01	0	0	1	+2.49e+03	+1.11e+02	0	1
HS100LNP	7	2	16	2	+6.89e+02	+1.74e+01	3	39	5	+7.17e+02	+1.97e+01	13	51
HS111LNP	10	3	9	1	-4.78e+01	+4.91e-06	2	0	10	-4.62e+01	+7.49e-01	10	1
HS27	3	1	2	0	+8.77e+01	+2.03e+02	3	5	1	+2.54e+01	+1.41e+02	11	34
GENHS28	10	8	1	0	+9.27e-01	+5.83e+01	0	0	1	+2.49e+03	+1.11e+02	0	1
HS39	4	2	11	1	-1.00e+00	+8.56e-05	1	0	10	-9.69e-01	+2.26e-01	5	1
HS40	4	3	4	0	-2.50e-01	+1.95e-06	0	0	3	-2.49e-01	+3.35e-02	2	1
HS42	4	2	4	1	+1.39e+01	+3.94e-04	1	0	1	+1.50e+01	+2.00e+00	3	1
HS52	5	3	1	0	+5.33e+00	+1.90e+02	1	0	1	+8.65e+03	+3.86e+02	0	1
HS6	2	1	1	0	+4.84e+00	+1.56e+00	32	136	1	+4.84e+00	+1.56e+00	32	136
HS61	3	2	1	0	-5.88e+01	+2.40e+01	INF	INF	1	-5.88e+01	+2.40e+01	INF	INF
HS7	2	1	7	1	-2.35e-01	+1.18e+00	7	2	8	+3.79e-01	+1.07e+00	5	2
HS77	5	2	13	30	+2.42e-01	+1.26e-02	0	0	17	+5.52e+02	+4.54e+02	3	11
HS78	5	3	6	0	-2.92e+00	+3.65e-04	1	0	10	-1.79e+00	+1.77e+00	2	30
HS79	5	3	13	21	+7.88e-02	+5.51e-02	0	2	10	+9.70e+01	+1.21e+02	0	24
LINCONT	249	419	1	0	+0.00e+00	+0.00e+00	INF	INF	1	+0.00e+00	+0.00e+00	INF	INF
LUKVLE1	10000	9998	7	5	+7.87e-01	+4.18e-04	0	0	19	+1.09e+09	+4.53e+07	0	238
LUKVLE3	10000	2	9	7	+2.50e+00	+8.24e-01	0	5	12	+8.68e+06	+5.85e+05	0	79
MARATOS	2	1	4	0	-1.00e+00	+8.59e-05	1	0	3	-9.96e-01	+9.02e-02	2	1
MWRIGHT	5	3	17	6	+2.31e+01	+5.78e-05	1	0	7	+5.07e+01	+1.04e+01	12	20
OPTCTRL3	4499	3000	1	9	+3.51e+01	+1.83e+01	0	15	4	+1.15e+05	+4.57e+03	1	21
OPTCTRL6	4499	3000	1	9	+3.51e+01	+1.83e+01	0	15	4	+1.15e+05	+4.57e+03	1	21
ORTHREGB	27	6	10	13	+1.79e-04	+2.71e+06	0	31	10	+2.73e+00	+1.60e+00	0	10

The results we obtained are provided in Table 2. For each problem, we indicate the number of variables (n), number of equality constraints (m), number of V-ITERATIONS (#V), number of F-ITERATIONS (#F), objective function value at the end of phase 1 (f), and dual infeasibility value at the end of phase 1 ($\|g + J^Ty\|$).

We write INF to indicate that an algorithm did not run phase 2 due to the fact that phase 1 ended with an infeasible stationary point (i.e., the second condition in (54) was satisfied). The results illustrate that, within a comparable number of iterations, TF typically yields better final points from phase 1. This can be seen in the fact that the objective at the end of phase 1, dual infeasibility at the end of phase 1, and the number of iterations required in phase 2 are all typically smaller for TF than they are for TF-V-ONLY. Note that for some problems, such as BT1, TF only performs V-ITERATIONS in phase 1, yet yields a better final point than does TF-V-ONLY; this occurs since the phase 1 iterations in TF may involve nonzero tangential steps.

6. Conclusion. An algorithm has been proposed for solving equality constrained optimization problems. Based on trust funnel and trust region ideas from [17, 23], the algorithm represents a next step toward the design of practical methods for solving constrained optimization problems that offer strong worst-case iteration complexity properties. In particular, the algorithm involves two phases, the first seeking (approximate) feasibility and the second seeking optimality, where a key contribution is the fact that improvement in the objective function is sought in both phases.

REFERENCES

[1] L. T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, eds., *Real-Time PDE-Constrained Optimization*, Comput. Sci. Eng. 3, SIAM, Philadelphia, PA, 2007.

[2] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, *Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models*, SIAM J. Optim., 26 (2016), pp. 951–967, https://doi.org/10.1137/15M1031631.

[3] G. Biros and O. Ghattas, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part* I*: The Krylov–Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[4] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact SQP method for equality constrained optimization*, SIAM J. Optim., 19 (2008), pp. 351–369.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* I*: Motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* II*: Worst-case function- and derivative-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.

[7] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming*, SIAM J. Optim., 21 (2011), pp. 1721–1739, https://doi.org/10.1137/11082381X.

[8] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Optimal Newton-Type Methods for Nonconvex Smooth Optimization Problems*, Technical report, ERGO 11-009, University of Edinburgh, Edinburgh, 2011.

[9] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity*, IMA J. Numer. Anal., 32 (2012), pp. 1662–1695.

[10] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization*, SIAM J. Optim., 23 (2013), pp. 1553–1574.

[11] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *On the complexity of finding first-order critical points in constrained nonlinear optimization*, Math. Program., 144 (2014), pp. 93–106.

[12] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, MOS-SIAM Ser. Optim. 1, SIAM, Philadelphia, PA, 2000.

[13] L. Cui, W. Kuo, H. T. Loh, and M. Xie, *Optimal allocation of minimal & perfect repairs under resource constraints*, IEEE Transactions on Reliability, 53 (2004), pp. 193–199.

[14] F. E. Curtis, N. I. M. Gould, D. P. Robinson, and Ph. L. Toint, *An interior-point trust-funnel algorithm for nonlinear optimization*, Math. Program., 161 (2016), pp. 73–134, https://doi.org/10.1007/s10107-016-1003-9.

[15] F. E. Curtis, T. C. Johnson, D. P. Robinson, and A. Wächter, *An inexact sequential quadratic optimization algorithm for nonlinear optimization*, SIAM J. Optim., 24 (2014), pp. 1041–1074.

[16] F. E. Curtis, J. Nocedal, and A. Wächter, *A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians*, SIAM J. Optim., 20 (2009), pp. 1224–1249.

[17] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust region algorithm with a worst-case iteration complexity of* $\mathcal{O}(\epsilon^{-3/2})$ *for nonconvex optimization*, Math. Program., 162 (2017), pp. 1–32, https://doi.org/10.1007/s10107-016-1026-2.

[18] F. E. Curtis, D. P. Robinson, and M. Samadi, *Complexity Analysis of a Trust Funnel Algorithm for Equality Constrained Optimization*, Technical report, 16T-013, Lehigh University, Bethlehem, PA, 2017.

[19] N. I. M. Gould, D. Orban, and Ph. Toint, *CUTEst: A Constrained and Unconstrained Testing Environment With Safe Threads*, Technical report, RAL-TR-2013-005, Rutherford Appleton Laboratory, Didcot, 2013.

[20] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: Global convergence*, SIAM J. Optim., 20 (2010), pp. 2023–2048.

[21] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: Local convergence and practical issues*, SIAM J. Optim., 20 (2010), pp. 2049–2079.

[22] N. I. M. Gould and D. P. Robinson, *A second-derivative SQP method with a "trust-region-free" predictor step*, IMA J. Numer. Anal., 32 (2012), pp. 580–601.

[23] N. I. M. Gould and Ph. L. Toint, *Nonlinear programming without a penalty function or a filter*, Math. Program., 122 (2008), pp. 155–196, https://doi.org/10.1007/s10107-008-0244-7.

[24] A. Griewank, *The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms*, Technical report, NA/12, University of Cambridge, Cambridge, 1981.

[25] R. J. Hathaway, *A constrained formulation of maximum-likelihood estimation for normal mixture distributions*, Ann. Statist., 13 (1985), pp. 795–800.

[26] N.-S. Hsu and K.-W. Cheng, *Network flow optimization model for basin-scale water supply planning*, J. Wat. Resour. Plann. Manag., 128 (2002), pp. 102–112.

[27] P. Marti, C. Lin, S. A. Brandt, M. Velasco, and J. M. Fuertes, *Optimal state feedback based resource allocation for resource-constrained control tasks*, in Proceedings of the 25th IEEE International Real-Time Systems Symposium, IEEE Press, Piscataway, NJ, 2004, pp. 161–172.

[28] J. M. Martínez, *On high-order model regularization for constrained optimization*, SIAM J. Optim., 27 (2017), pp. 2447–2458, https://doi.org/10.1137/17M1115472.

[29] J. L. Morales, J. Nocedal, and Y. Wu, *A sequential quadratic programming algorithm with an additional equality constrained phase*, IMA J. Numer. Anal., 32 (2012), pp. 553–579.

[30] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. and Stat. Comput., 4 (1983), pp. 553–572.

[31] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 117–205.

[32] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.

[33] K. E. Nygard, P. R. Chandler, and M. Pachter, *Dynamic network flow optimization models for air vehicle resource allocation*, in Proceedings of the 2001 American Control Conference, Vol. 3, IEEE Press, Piscataway, NJ, 2001, pp. 1853–1858.

[34] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.

[35] M. Weiser, P. Deuflhard, and B. Erdmann, *Affine conjugate adaptive Newton methods for nonlinear elastomechanics*, Optim. Methods Softw., 22 (2007), pp. 413–431.