A MATRIX-FREE ALGORITHM FOR EQUALITY CONSTRAINED OPTIMIZATION PROBLEMS WITH RANK-DEFICIENT JACOBIANS

FRANK E. CURTIS*, JORGE NOCEDAL†, AND ANDREAS WÄCHTER‡

Abstract. We present a line search algorithm for large-scale equality constrained optimization designed primarily for problems with (near) rank-deficient Jacobian matrices. The method is matrix-free (i.e., it does not require explicit representations or factorizations of derivative matrices), allows for inexact step computations, and does not require inertia information in order to solve nonconvex problems. The main components of the approach are a trust region subproblem for handling ill-conditioned or inconsistent linear models of the constraints and a process for attaining a sufficient reduction in a local model of a penalty function for the complete primal-dual step. We show that the algorithm is globally convergent to first-order optimal points or to stationary points of a feasibility measure. Numerical results are presented.

Key words. large-scale optimization, constrained optimization, nonconvex programming, trust regions, inexact linear system solvers, Krylov subspace methods

AMS subject classifications. 49M05, 49M37, 65K05, 90C06, 90C26, 90C30

1. Introduction.

A variety of sophisticated algorithms exist for the solution of large-scale nonlinear optimization problems. They benefit from both global and fast local convergence guarantees and have shown in practice to require few iterations and function/derivative evaluations in order to locate a local solution point. Some of these algorithms are also equipped to cope with rank-deficient or ill-conditioned Jacobian matrices. In some cases this is done by regularizing the constraints via a penalty function [6, 9, 10, 27], and in others it is accomplished by employing trust region techniques [2, 7, 16, 21]. A major drawback of most of these approaches, however, is that they require explicit representations and factorizations of large iteration matrices throughout the solution process, which makes their use impractical for large classes of problems.

In this paper, we develop and analyze an algorithm for equality constrained problems of the form

$$\min_{x\in\mathbb{R}^n} \; f(x) \;\; \text{s.t. } c(x) = 0, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are sufficiently smooth functions. We are interested in very large-scale applications for which contemporary methods cannot be employed, and in particular those where the constraint functions may be ill-conditioned or even inconsistent. This includes many problems in which the constraint functions are given as a discretized set of partial differential equations (PDEs). Our goal is to extend tools from the wealth of optimization research in order to define a robust *matrix-free* technique for the solution of these types of problems.

We categorize an optimization algorithm as matrix-free if the approach does not require the explicit formation or factorization of any matrix. Fortunately, as can be

[*] Courant Institute of Mathematical Sciences, New York University, New York, NY, USA. This author was supported by National Science Foundation grant DMS 0602235.

[†] Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. This author was supported by National Science Foundation grant CCF-0514772 and by Department of Energy grant DE-FG02-87ER25047-A004.

[‡] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA.

seen in the algorithms proposed in [3, 4, 13, 22, 25], such computationally expensive tasks can be avoided by considering iterative solution procedures for the step computations of the algorithm. The storage requirements for these techniques are often drastically less than that for direct factorization methods, while the computational costs essentially reduce to matrix-vector products with the Jacobian of the constraint functions, its transpose, and the (approximate) Hessian of the Lagrangian function

$$\mathcal{L}(x,\lambda) \triangleq f(x) + \lambda^T c(x),$$

all of which are viable operations for many large-scale problems. The difficulty, however, is that traditional algorithmic challenges normally faced in nonlinear programming, such as the efficient handling of nonconvexity of the problem functions, rank deficiency of the constraint Jacobians, and the achievement of global and local convergence guarantees, become greater when the algorithm is limited to these types of operations.

The goal of this paper is thus to design and analyze an efficient matrix-free algorithm for problem (1.1) with the following properties. First, given an instance of problem (1.1) and an arbitrary starting point, we would like our method to be globally convergent to first-order optimal points, or at least to infeasible stationary points of the feasibility measure

$$\varphi(x) \triangleq \|c(x)\|_2.$$

If f and c are first-order differentiable, we can define $g(x)$ as the gradient of $f(x)$ and $A(x)$ as the Jacobian of $c(x)$ to state the first-order optimality conditions of problem (1.1) as

$$\nabla\mathcal{L}(x,\lambda) = \begin{bmatrix} g(x) + A(x)^T\lambda \\ c(x) \end{bmatrix} = 0, \tag{1.2}$$

for some vector of Lagrange multipliers $\lambda \in \mathbb{R}^t$. An infeasible stationary point of φ, on the other hand, can be classified as one satisfying

$$\|c(x)\|_2 > 0 \ \text{ and } \ \nabla\varphi(x) = \frac{A(x)^T c(x)}{\|c(x)\|_2} = 0. \tag{1.3}$$

If possible, we would also like our method to emulate the fast local convergence behavior of Newton's method applied to the optimality conditions (1.2). With $c^i(x)$ and λ^i denoting the ith constraint function and its corresponding dual variable, respectively, we define

$$W(x,\lambda) \triangleq \nabla^2_{xx}\mathcal{L}(x,\lambda) = \nabla^2_{xx} f(x) + \sum_{i=1}^{t} \lambda^i \nabla^2_{xx} c^i(x)$$

as the Hessian of the Lagrangian at (x,λ). Then, a Newton iteration from an iterate (x_k,λ_k) has the form of a linear system of equations (e.g., see [15]):

$$\begin{bmatrix} W(x_k,\lambda_k) & A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = -\begin{bmatrix} g(x_k) + A(x_k)^T\lambda_k \\ c(x_k) \end{bmatrix}. \tag{1.4}$$

When $W(x_k,\lambda_k)$ is positive definite on the null space of $A(x_k)$, the primal step component d_k corresponds to the unique solution to the sequential quadratic programming

(SQP) subproblem

$$\begin{aligned}\min_{d\in\mathbb{R}^n}\ & f(x_k)+g(x_k)^Td+\tfrac{1}{2}d^TW(x_k,\lambda_k)d\\ \text{s.t. } & c(x_k)+A(x_k)d_k=0.\end{aligned}\tag{1.5}$$

Unfortunately, without explicit representations of $W(x_k,\lambda_k)$ and $A(x_k)$, an exact solution to (1.4) is not easily attained, and when $W(x_k,\lambda_k)$ is not positive definite over the null space of $A(x_k)$ we may no longer characterize solutions to (1.4) as those of (1.5). Moreover, many of the safeguards implemented in contemporary optimization software to cope with this latter difficulty are again unavailable or impractical in a matrix-free environment.

The algorithm in this paper overcomes these obstacles by extending the inexact Newton algorithms developed by Byrd, Curtis, and Nocedal for convex problems in [4] and for nonconvex problems in [3]. The defining characteristic of these approaches is that the primal-dual step computation is performed via an iterative procedure for solving the primal-dual equations (1.4), where inexact solutions are accepted or rejected based on the reductions they produce in a local model of the penalty function

$$\phi(x;\pi)\triangleq f(x)+\pi\|c(x)\|_2.$$

Here, $\pi>0$ is a penalty parameter that is updated automatically during the solution process. The model about x_k has the form

$$m_k(d;\pi)\triangleq f(x_k)+g(x_k)^Td+\pi\|c(x_k)+A(x_k)d\|_2$$

and the reductions obtained can be computed easily for any trial step d_k as

$$\begin{aligned}\Delta m_k(d_k;\pi)\triangleq\ & m_k(0;\pi)-m_k(d_k;\pi)\\ =\ & -g(x_k)^Td_k-\pi(\|c(x_k)\|_2-\|c(x_k)+A(x_k)d_k\|_2).\end{aligned}\tag{1.6}$$

If a sufficiently accurate solution to the system (1.4) yields a sufficiently large value for $\Delta m_k(d_k;\pi_k)$ for an appropriate value of π_k, then the (inexact) solution (d_k,δ_k) to (1.4) can be considered an acceptable search direction and the iterative step computation can be terminated.

The main challenge in this work, to extend the methods in [3, 4], is that we do not assume that the constraint Jacobians $\{A_k\}$ have full row rank throughout the solution process. We also guarantee that the algorithm remains well-defined when applied to an instance of problem (1.1) that is locally infeasible. The approach is still centered on the achievement of sufficient reductions in m_k during each iteration, though we can now no longer rely exclusively on the Newton equations (1.4), since this system may be ill-conditioned or even inconsistent during a particular iteration.

The organization of this paper is as follows. In §2 we motivate and present our matrix-free approach for equality constrained optimization. The global behavior of the approach is the topic of §3. Results from some preliminary numerical experiments are presented in §4, and concluding remarks are provided in §5.

Notation. All norms are considered Euclidean (or ℓ_2) unless otherwise indicated. We drop functional notation and use subscripts to denote iteration information for functions as with variables; i.e., $A_k\triangleq A(x_k)$ and similarly for other quantities. We use the expression $M_1\succ M_2$ to indicate that the matrix M_1-M_2 is positive definite.

2. A Matrix-free Primal-Dual Method. We begin our motivation for the particulars of our approach by briefly summarizing the methodology developed in [3] for nonconvex optimization. The central tenet of this approach is that a sufficiently accurate solution (d_k, δ_k) to the primal-dual equations (1.4) is an acceptable search direction provided that the reduction obtained in the model m_k satisfies

$$\Delta m_k(d_k;\pi_k) \geq \max\{\tfrac{1}{2}d_k^T W_k d_k, \theta\|d_k\|^2\} + \sigma\pi_k \max\{\|c_k\|, \|c_k + A_k d_k\| - \|c_k\|\} \tag{2.1}$$

for given constants $0 < \sigma < 1$ and $\theta > 0$ and an appropriate π_k. (We have simplied (2.1) slightly from [3] for ease of exposition.) If W_k is sufficiently positive definite and a reduction in linear infeasibility of the constraints is obtained, then (2.1) simplifies (the first terms in the max expressions dominate) and only requires that the reduction in m_k is sufficiently large with respect to the constraint feasibility measure and a quadratic term corresponding to the objective. However, since W_k may not be positive definite and an inexact solution may not yield a constraint reduction, the expression has been reinforced so that the right-hand-side of (2.1) remains sufficiently positive for each potential step d_k. We may increase the penalty parameter π_k in order to satisfy (2.1) for a given d_k, but only under certain conditions including, naturally, a condition that a sufficiently large constraint model decrease has been attained.

If the singular values of $\{A_k\}$ are bounded away from zero over all k, then the procedure in [3] yields global convergence guarantees under common conditions. If this is not the case, however, then such an approach needs to be reinforced to be well-defined and to guarantee progress toward worthwhile regions of the search space.

We are now ready to describe our algorithm, which is composed of two main stages. Let us briefly summarize the approach here before describing its details below. First, a *normal* step is computed as a move toward the satisfaction of a linear model of the constraint functions from the current iterate. The subproblem is formulated within a trust region to implicitly regularize the model and ensure that the step remains within a local region of the search space. Second, a *tangential* component and a displacement for the Lagrange multipliers are computed via an adapted system of primal-dual equations as moves toward optimality and dual feasibility, respectively. The total step in the primal space is then the concatenation of the normal and tangential components. Under certain conditions, the full primal-dual step will correspond to a good approximate solution of (1.4), and of the SQP subproblem (1.5) in convex regions. Otherwise, the safeguards embedded in the step computation will at least provide a sufficient reduction in m_k in a manner similar to the steps computed in the algorithms in [3, 4], or will yield progress in attaining dual feasibility.

We begin with the computation of a normal step component. Here, we consider the trust region subproblem

$$\begin{aligned} \min_{v\in\mathbb{R}^n}\ & \tfrac{1}{2}\|c_k + A_k v\|^2 \\ \text{s.t. } & \|v\| \leq \omega\|A_k^T c_k\| \end{aligned} \tag{2.2}$$

for a given constant $\omega > 0$. An efficient method for computing an approximate solution to this problem in our context of matrix-free optimization is the conjugate gradient (CG) method with Steihaug stop tests; see §4 and [24] for more information. In general, however, we simply assume that v_k lies in the range space of A_k^T and that the usual Cauchy decrease condition

$$\|c_k\| - \|c_k + A_k v_k\| \geq \gamma(\|c_k\| - \|c_k + \alpha_k^c A_k v_k^c\|) \tag{2.3}$$

holds for some $\gamma \in (0,1]$, where $v_k^c \triangleq -A_k^T c_k$ is the steepest descent direction for the objective of problem (2.2) and α_k^c solves

$$\begin{aligned} \min_{\alpha^c \geq 0} \ & \tfrac{1}{2}\|c_k + \alpha^c A_k v_k^c\|^2 \\ \text{s.t. } & \|\alpha^c A_k^T c_k\| \leq \omega \|A_k^T c_k\|. \end{aligned} \tag{2.4}$$

Since $\alpha_k^c = 0$ is a feasible solution to (2.4) it follows that

$$\|c_k\| - \|c_k + \alpha_k^c A_k v_k^c\| \geq 0, \tag{2.5}$$

and it is clear that (2.3) is satisfied by an exact solution to problem (2.2).

Having computed a normal step v_k, we modify the primal-dual system (1.4) to

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ -A_k v_k \end{bmatrix}, \tag{2.6}$$

so that with an exact solution to this system the tangential component

$$u_k \triangleq d_k - v_k \tag{2.7}$$

will lie in the null space of A_k (i.e., an exact solution will yield $A_k u_k = 0$) and the reduction obtained by d_k in the linear model of the constraints will be equivalent to that obtained by v_k. If W_k is positive definite in the null space of A_k, then an exact solution of (2.6) yields a solution to a perturbed version of the SQP subproblem (1.5); namely, d_k and u_k solve

$$\left\{ \begin{array}{l} \min\limits_{d \in \mathbb{R}^n} g_k^T d + \frac{1}{2} d^T W_k d \\ \text{ s.t. } A_k d = A_k v_k \end{array} \right\} \quad \text{and} \quad \left\{ \begin{array}{l} \min\limits_{u \in \mathbb{R}^n} (g_k + W_k v_k)^T u + \frac{1}{2} u^T W_k u \\ \text{ s.t. } A_k u_k = 0 \end{array} \right\}, \tag{2.8}$$

respectively.

In an inexact environment, an iterative procedure is employed to solve (2.6), which during each iteration yields (d_k, δ_k) corresponding to the residual vector

$$\begin{bmatrix} \rho_k \\ r_k \end{bmatrix} \triangleq \begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} + \begin{bmatrix} g_k + A_k^T \lambda_k \\ -A_k v_k \end{bmatrix}. \tag{2.9}$$

For each trial search direction (d_k, δ_k) with $(\rho_k, r_k) \neq 0$, we cannot be sure that u_k lies in the null space of the constraint Jacobian A_k or that it sufficiently approximates a solution to problem (2.8). We are thus confronted with the challenge of determining appropriate conditions that can be used to detect when a given inexact solution is an acceptable search direction.

We begin to state our conditions for these purposes by describing some necessary characteristics of the tangential component u_k of the primal step d_k.

Tangential component condition. *A component u_k must satisfy*

$$\|u_k\| \leq \psi \|v_k\| \tag{2.10}$$

or

$$\tfrac{1}{2} u_k^T W_k u_k \geq \theta \|u_k\|^2 \tag{2.11a}$$

$$\textit{and} \quad (g_k + W_k v_k)^T u_k + \tfrac{1}{2} u_k^T W_k u_k \leq \zeta \|v_k\|, \tag{2.11b}$$

where $\theta > 0$ and $\psi, \zeta \geq 0$ are given constants.

The primary purpose of the tangential component condition (2.10)/(2.11) is to exert sufficient control on the norm or on the optimality properties (with respect to (2.8)) of the tangential component u_k. Equation (2.10) states that the step may be acceptable provided that it is contained in a region proportional in size to the length of the normal component v_k. Alternatively, the tangential component may be acceptable provided it yields a low enough objective value to problem (2.8). Since $u_k = 0$ is feasible to problem (2.8), the optimal objective value is at most zero, and we note that if $W_k \succ 2\theta I$, then only (2.11b) need be enforced.

We also enforce the following condition for each nonzero step in the primal space, which can be seen as an adapted version of (2.1).

Model reduction condition. *A step (d_k, δ_k) with $d_k \neq 0$ must satisfy*

$$\Delta m_k(d_k; \pi_k) \geq \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|) \tag{2.12}$$

for some $0 < \sigma < 1$, $\pi_k > 0$, and θ defined in the tangential component condition.

There are two important differences between the model reduction condition (2.12) and (2.1). First, we replace d_k with u_k in the first term on the right-hand-side. In fact, the algorithm in [3] may benefit by this replacement, but in that method an explicit tangential component is not computed. Second, the fact that we compute the step components v_k and u_k explicitly and the fact that v_k satisfies (2.3) allow us to simplify the last term on the right-hand-side related to the deviation from linearized feasibility.

We are now ready to define three termination tests for an iterative solver applied to the primal-dual system (2.6). Once one of these tests is satisfied, we may terminate the linear solver as we have computed an acceptable search direction. We refer to these tests as sufficient merit function approximation reduction termination tests (SMART tests for short) as in [3, 4]. Each requires that the tangential component condition (2.10) or (2.11) is satisfied, ensures that the model reduction condition (2.12) is satisfied for an appropriate value of the penalty parameter π or that an appropriate step in the dual space is obtained, and poses appropriate bounds on the residual vectors ρ_k and r_k. Motivation for most of the quantities expressed in these tests can be found in [3, 4].

The first two termination tests enforce the following requirement on the residual vector:

$$\|\rho_k\| \leq \kappa \min\left\{\left\|\begin{bmatrix} g_k + A_k^T\lambda_k \\ A_k v_k\end{bmatrix}\right\|, \left\|\begin{bmatrix} g_{k-1} + A_{k-1}^T\lambda_k \\ A_{k-1} v_{k-1}\end{bmatrix}\right\|\right\}, \tag{2.13}$$

for some $\kappa \in (0, 1)$. The inequality (2.13) is similar to bounds commonly found in inexact Newton techniques, with two notable differences. First, we disregard the residual r_k as this quantity is controlled implicitly by the tangential component condition, even though in practical implementations explicit control of r_k may be beneficial to promote fast local convergence. Second, the second term in the min expression on the right-hand-side is necessary for ensuring the limits (3.8) and (3.10) corresponding to convergence toward dual feasibility (see Lemma 3.19).

In addition to enforcing (2.13), steps satisfying the first test termination correspond to a productive step in the primal space for the most recent value of the penalty parameter.

Termination test 1. *A search direction* (d_k, δ_k) *is acceptable if the tangential component condition* (2.10) *or* (2.11) *is satisfied, the model reduction condition* (2.12) *is satisfied for a given* $0 < \sigma < 1$ *and* $\pi_k = \pi_{k-1}$*, and if for* $\kappa = \kappa_a \in (0,1)$ *the residual bound* (2.13) *holds.*

As steps satisfying this test satisfy the model reduction condition (2.12) for the most recent value of the penalty parameter, we maintain $\pi_k \leftarrow \pi_{k-1}$ during any iteration k when this test is satisfied.

Steps satisfying the second termination test correspond to productive perturbations in the Lagrange multiplier estimates. Inclusion of this test is necessary for the particular cases when x_k is a feasible and optimal solution to problem (1.1), or is an infeasible stationary point to the feasibility measure φ, but $g_k + A_k^T\lambda_k \neq 0$, as in these situations we may otherwise require an exact (or near exact) solution to the primal-dual system (2.6) to produce an acceptable step. The test may also save computational expenses when x_k is in the neighborhood of such points.

Termination test 2. *If for a given constant* $\epsilon_b > 0$ *we have*

$$\|A_k^T c_k\| \leq \epsilon_b \|g_k + A_k^T\lambda_k\|, \tag{2.14}$$

then a step $(d_k, \delta_k) \leftarrow (0, \delta_k)$ *is acceptable if for* $\kappa = \kappa_b \in (0,1)$ *the residual bound* (2.13) *holds; i.e.,*

$$\left\|g_k + A_k^T(\lambda_k + \delta_k)\right\| \leq \kappa_b \min\left\{\left\|g_k + A_k^T\lambda_k\right\|, \left\|\begin{bmatrix} g_{k-1} + A_{k-1}^T\lambda_k \\ A_{k-1}v_{k-1}\end{bmatrix}\right\|\right\}. \tag{2.15}$$

We restrict consideration of this test to iterations where (2.14) is satisfied so that the algorithm does not only reduce dual infeasibility. Moreover, it is important to note that for accepted steps satisfying only this test we maintain $\pi_k \leftarrow \pi_{k-1}$ and set $x_{k+1} \leftarrow x_k$ (i.e., we reset $v_k \leftarrow 0$ and $u_k \leftarrow 0$ to get $d_k \leftarrow 0$), while only perturbing the multiplier estimates.

The last termination test corresponds to a sufficient reduction in the linear model of constraint infeasibility and enforces appropriate conditions under which we may consider an increase in the penalty parameter value. For this test to be considered during iteration k, we require that $\|c_k\| - \|c_k + A_k v_k\| > 0$.

Termination test 3. *A step* (d_k, δ_k) *is acceptable if the tangential component condition* (2.10) *or* (2.11) *is satisfied and if for given constants* $0 < \epsilon < 1$ *and* $\beta > 0$ *we have*

$$\begin{aligned} \|c_k\| - \|c_k + A_k d_k\| \geq\ & \epsilon(\|c_k\| - \|c_k + A_k v_k\|) > 0, && \text{(2.16a)} \\ \textit{and} \quad \|\rho_k\| \leq\ & \beta(\|c_k\| - \|c_k + A_k v_k\|). && \text{(2.16b)} \end{aligned}$$

Notice that (2.16a) requires the reduction in the constraint model to be at least a proportion of that obtained by v_k itself, but no more (since no more can be expected based on the form of (2.6)). For steps satisfying termination test 3, we set

$$\pi_k \geq \frac{g_k^T d_k + \max\{\frac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\}}{(1-\tau)(\|c_k\| - \|c_k + A_k d_k\|)} \triangleq \pi_k^{trial}, \tag{2.17}$$

which implies (see (1.6))

$$\begin{aligned}\Delta m_k(d_k;\pi_k) \geq\ & \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \tau\pi_k(\|c_k\| - \|c_k + A_k d_k\|)\\ \geq\ & \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \tau\epsilon\pi_k(\|c_k\| - \|c_k + A_k v_k\|)\end{aligned}$$

and so the model reduction condition (2.12) is satisfied for $\sigma = \tau\epsilon$. From now on we assume the constants τ, σ, and ϵ are chosen to satisfy this relationship for consistency between termination tests 1 and 3.

Now, as in [3], we note that the (approximate) Hessian of the Lagrangian W_k may need to be modified during the solution process if the problem is not strictly convex. During the step computation, we call for such a modification according to the following rule.

Hessian modification strategy. *Let W_k be the current Hessian approximation and let a trial step (d_k, δ_k) be given. If d_k does not satisfy* (2.10) *or* (2.11a)*, then modify W_k; otherwise, maintain the current W_k.*

For the algorithm to be well-posed, we assume that after a finite number of modifications the matrix W_k has $W_k \succ 2\theta I$ for the constant θ defined in the tangential component condition.

Finally, once an acceptable step has been computed, we perform a backtracking line search on the penalty function $\phi(x;\pi_k)$. As shown in Lemma 3.9 below, the expression $-\Delta m_k(d_k;\pi_k)$ is an upper bound for the directional derivative of $\phi(x;\pi_k)$ along d_k from x_k. Since this latter quantity, denoted as $D\phi(d_k;\pi_k)$, is not easily computed, we choose α_k such that the Armijo condition

$$\phi(x_k + \alpha_k d_k;\pi_k) \leq \phi(x_k;\pi_k) - \eta\alpha_k\Delta m_k(d_k;\pi_k) \tag{2.18}$$

holds for a given $\eta \in (0,1)$. For the step δ_k in the dual space we compute a steplength coefficient α_k^λ so that $\lambda_{k+1} = \lambda_k + \alpha_k^\lambda\delta_k$ satisfies

$$\|g_k + A_k^T\lambda_{k+1}\| \leq \|g_k + A_k^T(\lambda_k + \delta_k)\|, \tag{2.19}$$

which can be done, for example, by solving

$$\min_{\alpha\in[0,1]} \ \tfrac{1}{2}\|g_k + A_k^T(\lambda_k + \alpha\delta_k)\|^2,$$

as is done in our implementation described in §4.

The complete algorithm is the following.

```
Algorithm TRINS: Trust Region Inexact Newton with SMART Tests
Choose parameters $0 < \kappa_a, \kappa_b, \epsilon, \tau, \eta < 1$ and $0 < \omega, \psi, \theta, \zeta, \epsilon_b, \beta, \delta_\pi$, and set $\sigma \leftarrow \tau\epsilon$
Initialize $x_0$, $\lambda_0$, and $\pi_{-1} > 0$, and define $v_{-1} \leftarrow 0$, $g_{-1} \leftarrow g_0$, and $A_{-1} \leftarrow A_0$
for $k = 0, 1, 2, \ldots$, until a termination test for (1.1) is satisfied
    Compute $f_k$, $g_k$, $c_k$, $A_k$, and $W_k$ and initialize $\pi_k \leftarrow \pi_{k-1}$
    Compute $v_k \in \mathrm{range}(A_k^T)$ as an approximate solution to (2.2) satisfying (2.3)
    Compute an approximate solution $(d_k, \delta_k)$ to (2.6)
    repeat
        Set $u_k \leftarrow d_k - v_k$
        if $(d_k, \delta_k)$ satisfies termination test 1 or 3, then break
        if $(d_k, \delta_k)$ satisfies termination test 2, then set $d_k \leftarrow 0$ and break
        Run the Hessian modification strategy to update $W_k$
        Compute an approximate solution $(d_k, \delta_k)$ to (2.6) with the current $W_k$
    endrepeat
    if termination test 3 is satisfied and (2.17) does not hold, set $\pi_k \leftarrow \pi_k^{trial} + \delta_\pi$
    if $d_k \neq 0$, compute the smallest $l \in \{0, 1, 2, \ldots\}$ so that $\alpha_k = 2^{-l}$ satisfies (2.18)
        else set $\alpha_k \leftarrow 1$
    Choose $\alpha_k^\lambda$ yielding (2.19)
    Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$ and $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k^\lambda \delta_k$
endfor
```

A description of a particular implementation of Algorithm TRINS is given in §4. We also provide guidelines for the selection of the many input parameters required by the algorithm.

3. Algorithm Analysis.

In this section we analyze the global behavior of Algorithm TRINS in the following setting.

Assumption 3.1. *The sequence $\{x_k, \lambda_k\}$ generated by Algorithm* TRINS *is contained in a convex set over which the functions f and c and their first derivatives are bounded. Moreover, the sequence $\{W_k\}$ is bounded over all k and the iterative linear system solver can solve* (2.6) *to an arbitrary accuracy for each original and modified W_k produced in the algorithm.*

We overload W_k to refer to the initial Hessian matrix used in iteration k and all subsequent perturbations formed via the Hessian modification strategy. For much of this section, however, W_k refers exclusively to the value for this matrix used to compute an acceptable step; i.e., the value for W_k computed after all modifications via the Hessian modification strategy have been performed.

There are a number of iterative linear solvers that have the properties required in Assumption 3.1 when the primal-dual system (2.6) is consistent, even if the primal-dual matrix is singular; see §4 and §5 for further discussion. Moreover, due to the structure of (2.6), inconsistency can only be caused by singularity of W_k. In such cases, the iterative solver will not converge, but further modifications of W_k as in the Hessian modification strategy will eventually produce a consistent system.

For convenience, we define the index sets T_1, T_2, and T_3 to denote iterations where termination test 1, 2, and 3 are satisfied, respectively.

3.1. Well-posedness of Algorithm TRINS.

Before analyzing the global behavior of our approach, it is important to verify that each iteration of Algorithm TRINS is well-defined under Assumption 3.1. If at iteration k we have

$$A_{k-1}^T c_{k-1} = 0 \quad \text{and} \quad g_{k-1} + A_{k-1}^T \lambda_k = 0, \quad \text{or} \tag{3.1a}$$

$$A_k^T c_k = 0 \quad \text{and} \quad g_k + A_k^T \lambda_k = 0, \tag{3.1b}$$

then we assume that the algorithm terminates finitely and returns (x_{k-1}, λ_k) or (x_k, λ_k), respectively. In such cases, the algorithm has arrived at a first-order optimal point (see (1.2)) or at least a stationary point of the feasibility measure φ (see (1.3)). The following lemma formalizes our discussion dealing of the remaining cases; i.e., iterations where neither (3.1a) nor (3.1b) hold.

Lemma 3.2. *If at iteration k neither* (3.1a) *nor* (3.1b) *holds, then the* `repeat` *loop of Algorithm* TRINS *is finite.*

Proof. Let j denote the iteration counter for the `repeat` loop of Algorithm TRINS; i.e., (d_k^j, δ_k^j), $j = 1, 2, \ldots$ denote the trial steps generated during iteration k. By our assumption that after a finite number of Hessian modifications we have $W_k \succ 2\theta I$, there exists an iteration j' such that $W_k = W_k^{j'}$ for all $j \geq j'$. That is, for all $j \geq j'$ either (2.10) or (2.11a) holds, which means

$$\|u_k^j\| \leq \psi\|v_k\| \quad \text{or} \tag{3.2a}$$
$$\tfrac{1}{2}(u_k^j)^T W_k u_k^j \geq \theta\|u_k^j\|^2. \tag{3.2b}$$

Further, under Assumption 3.1 we have

$$(\rho_k^j, r_k^j) \to 0. \tag{3.3}$$

We consider a series of cases and show that in each either termination test 1, 2, or 3 is satisfied for j sufficiently large.

We begin by showing that the tangential component condition is satisfied for all large j. First, if (3.2a) (or, equivalently, (2.10)) is satisfied for all large j, then the tangential component condition is satisfied for all large j. Otherwise, there is an infinite subsequence defined by $j \in J$ having

$$\|u_k^j\| > \psi\|v_k\|, \tag{3.4}$$

where with (3.2) we have that the inequality (3.2b) holds for all large $j \in J$. Let $\tilde{A}_k$ be the matrix obtained by removing all linearly dependent rows in A_k, and denote by $\tilde{r}_k^j$ the subvector of r_k^j with the corresponding entries removed. Since $\tilde{A}_k$ has the same range space as A_k, there exists $\tilde{\delta}_k^j$ for all j with

$$\tilde{A}_k^T \tilde{\delta}_k^j = A_k^T \delta_k^j = -g_k - A_k^T\lambda_k - W_k(v_k + u_k^j) + \rho_k^j.$$

Thus, for some γ_a, $\gamma_b > 0$, we have

$$\tilde{\delta}_k^j = \left[\tilde{A}_k\tilde{A}_k^T\right]^{-1}\tilde{A}_k\left(-g_k - A_k^T\lambda_k - W_k(v_k + u_k^j) + \rho_k^j\right) \leq \gamma_a + \gamma_b\|u_k^j\|$$

for all large $j \in J$, and with (3.2b) we find

$$\begin{aligned}
&(g_k + W_k v_k)^T u_k^j + \tfrac{1}{2}(u_k^j)^T W_k u_k^j \\
= \; & -\tfrac{1}{2}(u_k^j)^T W_k u_k^j - \lambda_k^T A_k u_k^j - (\tilde{\delta}_k^j)^T \tilde{A}_k u_k^j + (\rho_k^j)^T u_k^j \\
= \; & -\tfrac{1}{2}(u_k^j)^T W_k u_k^j - \lambda_k^T r_k^j - (\tilde{\delta}_k^j)^T \tilde{r}_k^j + (\rho_k^j)^T u_k^j \\
\leq \; & -\theta\|u_k^j\|^2 + \|\lambda_k\|\|r_k^j\| + (\gamma_a + \gamma_b\|u_k^j\|)\|\tilde{r}_k^j\| + \|\rho_k^j\|\|u_k^j\|.
\end{aligned} \tag{3.5}$$

Let us view the right-hand-sides of (3.5) for $j \in J$ as a sequence of concave quadratic functions of $\|u_k^j\|$. Each function in this sequence has the form

$$h^j(y) = -ay^2 + b^j y + c^j \tag{3.6}$$

with $a > 0$ and $\lim_{k\to\infty}\{b^j, c^j\} = 0$, and has the maximizer $b^j/2a$, which converges to zero as $j \to \infty$. Thus, for all large $j \in J$, the supremum of $h^j(y)$ subject to the constraint (3.4) occurs at $y^* = \psi\|v_k\|$ and yields the value

$$h^j(y^*) = -a\psi^2\|v_k\|^2 + b^j\psi\|v_k\| + c^j < 0.$$

Applying this strict inequality to (3.5), we have shown that (2.11) holds for all large $j \in J$. Thus, the tangential component condition is satisfied for all large j.

Next we consider the cases $\|A_k^T c_k\| > 0$ and $\|A_k^T c_k\| = 0$ and show that in each case the remaining conditions of termination test 1, 2, or 3 will eventually be satisfied. First, if $\|A_k c_k\| > 0$, then $\|c_k\| - \|c_k + A_k v_k\| > 0$ and $\|v_k\| > 0$ by (2.3) and (2.4). Therefore, the right-hand-side of (2.16b) is positive and it follows from (3.3) that (2.16b) is satisfied for all large j. Similarly, by (2.7), (2.9), and (3.3),

$$\begin{aligned}\frac{\|c_k\| - \|c_k + A_k d_k^j\|}{\|c_k\| - \|c_k + A_k v_k\|} &\geq \frac{\|c_k\| - \|c_k + A_k v_k\| - \|A_k u_k^j\|}{\|c_k\| - \|c_k + A_k v_k\|} \\ &= 1 - \frac{\|r_k^j\|}{\|c_k\| - \|c_k + A_k v_k\|} \geq \epsilon\end{aligned}$$

for all large j, where $\epsilon < 1$ in (2.16a). Thus, (2.16a) holds for all large j, and so all together we have shown that termination test 3 is eventually satisfied.

Now suppose $\|A_k^T c_k\| = 0$, which implies that $v_k = 0$ by the formulation of problem (2.2) and that $\|g_k + A_k^T \lambda_k\| \neq 0$ since (3.1b) does not hold. By (2.9) we then have $\|g_k + A_k^T(\lambda_k + \delta_k^j)\| = \|\rho_k^j - W_k u_k^j\|$. If $\liminf \|u_k^j\| = 0$, then by (3.3) this quantity is arbitrarily small for large j. This, along with the fact that the right-hand-side of (2.15) is nonzero as (3.1a) does not hold, implies that termination test 2 will eventually be satisfied. On the other hand, if $\|u_k^j\| \geq \gamma_c$ for some $\gamma_c > 0$ for all large j, then (3.2a) does not hold for all large j as $v_k = 0$. Therefore, (3.2b) holds for all large j and as before with the tangential component condition, we now find for the model reduction condition that for $v_k = 0$ with $\pi_k = \pi_{k-1}$ we have

$$\begin{aligned} & -\Delta m_k(d_k^j;\pi_k) + \max\{\tfrac{1}{2}(u_k^j)^T W_k u_k^j, \theta\|u_k^j\|^2\} + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|) \\ = {} & g_k^T d_k^j + \max\{\tfrac{1}{2}(u_k^j)^T W_k u_k^j, \theta\|u_k^j\|^2\} + \pi_k(\|c_k\| - \|c_k + A_k d_k^j\|) \\ = {} & g_k^T u_k^j + \tfrac{1}{2}(u_k^j)^T W_k u_k^j + \pi_k(\|c_k\| - \|c_k + r_k^j\|) \\ \leq {} & -\theta\|u_k^j\|^2 + \|\lambda_k\|\|r_k^j\| + (\gamma_a + \gamma_b\|u_k^j\|)\|\tilde{r}_k^j\| + \|\rho_k^j\|\|u_k^j\| + \pi_k\|r_k^j\|. \end{aligned} \tag{3.7}$$

The right-hand-side of (3.7) differs from that of (3.5) only in the term $\pi_k\|r_k^j\|$. Therefore, the right-hand-side of (3.7) can also be viewed as a concave quadratic function of the form (3.6), and as argued above it is negative for all large j, which means that the model reduction condition (2.12) is satisfied for all large j. Finally, by (3.3) and the fact that the right-hand-side of (2.13) is positive, the inequality (2.13) is satisfied for all large j, which implies that termination test 1 is eventually satisfied. □

We have thus shown that the iterative solution of the primal-dual equations (2.6) will produce a step (d_k, δ_k) satisfying termination test 1, 2, or 3.

Finally, we argue that the backtracking line search in Algorithm TRINS is guaranteed to produce an acceptable steplength coefficient α_k. By properties of the directional derivative $D\phi(d_k;\pi_k)$, it is clear that there exists an $\bar{\alpha}_k > 0$ such that the Armijo condition (2.18) with $-\Delta m_k(d_k;\pi_k)$ replaced by $D\phi(d_k;\pi_k)$ is satisfied for all $\alpha_k \in [0, \bar{\alpha}_k]$. Thus, since $D\phi(d_k;\pi_k) \leq -\Delta m_k(d_k;\pi_k)$ (see Lemma 3.8), the original condition (2.18) is also satisfied for all such α_k.

3.2. Global Convergence Analysis. The theorem we prove is the following.

Theorem 3.3. *If all limit points of $\{A_k\}$ have full row rank, then $\{\pi_k\}$ is bounded and*

$$\lim_{k\to\infty} \left\| \begin{bmatrix} g_k + A_k^T \lambda_{k+1} \\ c_k \end{bmatrix} \right\| = 0. \tag{3.8}$$

Otherwise,

$$\lim_{k\to\infty} \|A_k^T c_k\| = 0, \tag{3.9}$$

and if $\{\pi_k\}$ *is bounded then*

$$\lim_{k\to\infty} \|g_k + A_k^T \lambda_{k+1}\| = 0. \tag{3.10}$$

We prove Theorem 3.3 with a series of observations and lemmas.

This first result, related to a one-dimensional problem, will be useful for providing a tight bound on the reduction obtained in the constraint model by the normal step v_k. Its proof can be found in [5].

Lemma 3.4. *The optimal value* Φ^* *of the one-dimensional optimization problem*

$$\begin{aligned} &\min_{z\in\mathbb{R}} \ \Phi(z) \triangleq \tfrac{1}{2}z^2 a - zb \\ &\ s.t.\ z \le \Omega, \end{aligned}$$

where $b \ge 0$ *and* $\Omega > 0$*, satisfies*

$$\Phi^* \le -\tfrac{b}{2}\min\left\{\tfrac{b}{|a|}, \Omega\right\}.$$

Applying this result to problem (2.2) helps yield the following result.

Lemma 3.5. *There exists* $\gamma_1 > 0$ *such that, for all* $k \notin T_2$,

$$\|c_k\| - \|c_k + A_k v_k\| \ge \gamma_1 \frac{\|A_k^T c_k\|^2}{\|c_k\|} \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\} \triangleq \mathcal{C}_k. \tag{3.11}$$

Proof. Inequality (3.11) clearly holds when $\|A_k^T c_k\| = 0$ since then $\mathcal{C}_k = 0$ and the left-hand-side is nonnegative by (2.3) and (2.5). Thus, let us now assume that $\|A_k^T c_k\| \ne 0$.

According to problem (2.4), the quantity α_k^c solves

$$\begin{aligned} &\min_{\alpha^c \ge 0} \ \tfrac{1}{2}\left(\alpha^c\right)^2 \|A_k A_k^T c_k\|^2 - \alpha^c \|A_k^T c_k\|^2 \\ &\ \text{s.t. } \alpha^c \le \omega. \end{aligned}$$

Applying Lemma 3.4 yields

$$\begin{aligned} \tfrac{1}{2}\left(\|c_k + \alpha_k^c A_k v_k^c\|^2 - \|c_k\|^2\right) &= \tfrac{1}{2}\left(\alpha^c\right)^2 \|A_k A_k^T c_k\|^2 - \alpha^c \|A_k^T c_k\|^2 \\ &\le -\frac{1}{2}\|A_k^T c_k\|^2 \min\left\{\frac{\|A_k^T c_k\|^2}{\|A_k A_k^T c_k\|^2}, \omega\right\} \\ &\le -\frac{1}{2}\|A_k^T c_k\|^2 \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\}. \end{aligned}$$

Thus, since v_k satisfies the Cauchy decrease condition (2.3), we find with the relation $2a(a-b) \ge a^2 - b^2$ that

$$\begin{aligned} \|c_k\|(\|c_k\| - \|c_k + A_k v_k\|) &\ge \gamma \|c_k\|(\|c_k\| - \|c_k + \alpha_k^c A_k v_k^c\|) \\ &\ge \tfrac{1}{2}\gamma(\|c_k\|^2 - \|c_k + \alpha_k^c A_k v_k^c\|^2) \\ &\ge \frac{1}{2}\gamma \|A_k^T c_k\|^2 \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\}, \end{aligned}$$

which proves (3.11) since $\|c_k\| \neq 0$, as we are considering the case $\|A_k^T c_k\| \neq 0$. □

We can now present the following result that creates an envelope around the norm of the normal component v_k.

Lemma 3.6. *There exists $\gamma_2 > 0$ such that, for all $k \notin T_2$, we have*

$$\gamma_2 \|A_k^T c_k\|^2 \leq \|v_k\| \leq \omega \|A_k^T c_k\|, \tag{3.12}$$

and hence v_k is bounded in norm over all k.

Proof. The inequality on the right-hand-side of (3.12) follows from the formulation of problem (2.2). Thus, v_k is bounded in norm over all k since, under Assumption 3.1, the quantity $\|A_k^T c_k\|$ is bounded. The first inequality of (3.12) follows first from the triangle inequality, which yields

$$\|c_k\| - \|c_k + A_k v_k\| \leq \|A_k v_k\| \leq \|A_k\| \|v_k\|.$$

If $\|A_k\| = 0$, then (3.12) follows trivially. Otherwise, the above and (3.11) yield

$$\|v_k\| \geq \frac{\|c_k\| - \|c_k + A_k v_k\|}{\|A_k\|} \geq \gamma_1 \frac{\|A_k^T c_k\|^2}{\|A_k\| \|c_k\|} \min\left\{ \frac{1}{\|A_k^T A_k\|}, \omega \right\},$$

and so (3.12) follows from the fact that, under Assumption 3.1, $\|A_k\|$, $\|c_k\|$, and $\|A_k^T A_k\|$ are bounded over all k. □

This last result and the tangential component condition can now be used to bound the tangential components in norm.

Lemma 3.7. *The tangential components u_k are bounded in norm over all k.*

Proof. First, if $k \in T_2$, then $u_k = 0$, and if (2.10) is satisfied, then u_k is bounded by Lemma 3.6. The remaining case is when (2.11) is satisfied, which implies

$$(g_k + W_k v_k)^T u_k + \tfrac{1}{2} u_k^T W_k u_k \leq \zeta \|v_k\|,$$

and by (2.11a) we then have

$$\theta \|u_k\|^2 - \|u_k\| \|g_k + W_k v_k\| - \zeta \|v_k\| \leq 0.$$

The expression on the left-hand-side of this inequality is a convex quadratic function in $\|u_k\|$, and so the result follows from the fact that under Assumption 3.1 and by Lemma 3.6, all of the coefficients of this quadratic are bounded. □

We have thus shown that the entire primal step d_k is bounded in norm over all k.

Next, we turn to a series of results related to the model reduction condition and its connection to the directional derivative of the penalty function ϕ. The proof of this first result can be found in [4].

Lemma 3.8. *The directional derivative of the penalty function satisfies*

$$D\phi(d; \pi) \leq g^T d - \pi(\|c\| - \|c + Ad\|) = -\Delta m(d; \pi).$$

By applying the results above, this last lemma can be used to provide a more explicit bound for $D\phi(d_k; \pi_k)$.

Lemma 3.9. *There exists $\gamma_3 > 0$ such that, for all $k \notin T_2$, we have*

$$\Delta m_k(d_k; \pi_k) \geq \gamma_3 \left(\|u_k\|^2 + \pi_k \|A_k^T c_k\|^2 \right).$$

Proof. First note that, for all k, there exists $\gamma_4 > 0$ such that

$$\begin{aligned}\|A_k^T c_k\|^2 &= \left(\frac{\|c_k\|}{\gamma_1 \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\}}\right) \mathcal{C}_k \\ &= \frac{\|c_k\|}{\gamma_1} \max\left\{\|A_k^T A_k\|, \frac{1}{\omega}\right\} \mathcal{C}_k \\ &\leq \gamma_4 \mathcal{C}_k,\end{aligned}$$

since $\|c_k\|$ and $\|A_k^T A_k\|$ are bounded under Assumption 3.1. Then, for $k \notin T_2$, the model reduction condition (2.12) yields

$$\begin{aligned}\Delta m_k(d_k;\pi_k) \geq{}& \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|) \\ \geq{}& \theta\|u_k\|^2 + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|)\end{aligned}$$

and so by Lemma 3.5 we have

$$\begin{aligned}\Delta m_k(d_k;\pi_k) &\geq \theta\|u_k\|^2 + \sigma\pi_k\mathcal{C}_k \\ &\geq \theta\|u_k\|^2 + \tfrac{1}{\gamma_4}\sigma\pi_k\|A_k^T c_k\|^2.\end{aligned}$$

Thus, the result holds for $\gamma_3 = \min\{\theta, \frac{1}{\gamma_4}\sigma\} > 0$. □

The length of the total primal step d_k can be bounded above by a similar quantity.

Lemma 3.10. *There exists $\gamma_5 > 0$ such that, for all k, we have*

$$\|d_k\|^2 \leq \gamma_5\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^T c_k\|^2\right), \tag{3.13}$$

and hence

$$\|d_k\|^2 + \|A_k^T c_k\|^2 \leq 2\gamma_5\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^T c_k\|^2\right). \tag{3.14}$$

Proof. From (2.7) and (3.12) it follows that

$$\begin{aligned}\|d_k\|^2 &= \|u_k\|^2 + 2u_k^T v_k + \|v_k\|^2 \\ &\leq 2\left(\|u_k\|^2 + \|v_k\|^2\right) \\ &\leq 2\left(\|u_k\|^2 + \omega^2\|A_k^T c_k\|^2\right),\end{aligned}$$

so (3.13) holds for $\gamma_5 = 2\max\{1,\omega^2\}$. The inequality (3.14) follows trivially from (3.13). □

We are now ready to prove the limit (3.9). Here, it will be convenient to work with the scaled and shifted penalty function

$$\tilde{\phi}(x;\pi) \triangleq \tfrac{1}{\pi}\left(f(x) - f_{min}\right) + \|c(x)\|, \tag{3.15}$$

where f_{min} is the infimum of f over the convex set containing the iterates of the algorithm. This function has an interesting property that we first describe with the following lemma.

Lemma 3.11. *For all k,*

$$\tilde{\phi}(x_{k+1};\pi_{k+1}) \leq \tilde{\phi}(x_k;\pi_k) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k),$$

and so $\tilde{\phi}$ is monotonically decreasing.

Proof. First, if $k \in T_2$, then $d_k = 0$, $\Delta m_k(d_k;\pi_k) = 0$, and $\tilde{\phi}(x_{k+1};\pi_{k+1}) = \tilde{\phi}(x_k;\pi_k)$, and so the result follows trivially. Otherwise, by (2.18) it follows that

$$\tilde{\phi}(x_{k+1};\pi_k) \leq \tilde{\phi}(x_k;\pi_k) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k),$$

and so

$$\tilde{\phi}(x_{k+1};\pi_{k+1}) \leq \tilde{\phi}(x_k;\pi_k) + \left(\tfrac{1}{\pi_{k+1}} - \tfrac{1}{\pi_k}\right)(f_{k+1} - f_{min}) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k). \quad (3.16)$$

The fact that π_k is monotonically increasing, the nonnegativity of $(f_{k+1} - f_{min})$, and Lemma 3.9 then yields the result. □

The limit (3.9) now follows from the above results.

LEMMA 3.12. *The sequence $\{x_k\}$ yields the limit* (3.9).

Proof. Consider an arbitrary value $\gamma_6 > 0$ and define the set

$$S = \{x : \gamma_6 \leq \|A(x)^T c(x)\|\}.$$

Suppose that there exists $k' \geq 0$ such that $x_{k'} \in S$ and for all $k \geq k'$ we have $k \in T_2$. Then, $x_k = x_{k'}$ for all $k \geq k'$ and (2.15) and (2.19) yield

$$\|g_{k+1} + A_{k+1}^T\lambda_{k+1}\| \leq \|g_k + A_k^T(\lambda_k + \delta_k)\| \leq \kappa_b\|g_k + A_k^T\lambda_k\|.$$

Hence, $\|g_k + A_k^T\lambda_k\| \to 0$ as $\kappa_b < 1$. On the other hand, by the conditions of termination test 2 and the fact that $x_{k'} \in S$ we have

$$0 < \gamma_6 \leq \|A_{k'}^T c_{k'}\| = \|A_k^T c_k\| \leq \epsilon_\lambda\|g_k + A_k^T\lambda_k\|.$$

This contradiction shows that if $x_k \in S$, then there exists $k'' \geq k$ such that $x_{k''} \in S$ and $k'' \notin T_2$.

Now consider $x_k \in S$ with $k \notin T_2$. Then, Lemma 3.6 yields

$$\|v_k\| \geq \gamma_2\|A_k^T c_k\|^2 \geq \gamma_2\gamma_6^2.$$

Similarly, by Lemma 3.7 we may define

$$u_{\sup}^S \triangleq \sup\{\|u_l\| : x_l \in S\} < \infty,$$

where it follows that

$$\|u_k\| \leq \left(u_{\sup}^S/(\gamma_2\gamma_6^2)\right)\|v_k\|. \quad (3.17)$$

By Lemmas 3.6 and 3.9, there then exists a constant $\gamma_7 > 0$ such that

$$\begin{aligned}\Delta m_k(d_k;\pi_k) &\geq \gamma_3\pi_k\|A_k^T c_k\|^2 \\ &\geq \gamma_3\pi_k\frac{1}{\omega^2}\|v_k\|^2 \\ &\geq \gamma_7\pi_k\|d_k\|^2. \quad (3.18)\end{aligned}$$

Next, we note that if the line search condition (2.18) does not hold for some $\bar{\alpha} > 0$ during iteration k, then

$$\phi(x_k + \bar{\alpha}d_k;\pi_k) - \phi(x_k;\pi_k) > -\eta\bar{\alpha}\Delta m_k(d_k;\pi_k). \quad (3.19)$$

However, with Lemma 3.8, a Taylor expansion of $\phi(x;\pi_k)$ at x_k along d_k yields for some $\gamma_8 > 0$ independent of x_k

$$\begin{aligned}\phi(x_k+\bar\alpha d_k;\pi_k) - \phi(x_k;\pi_k) &\le \bar\alpha D\phi(d_k;\pi_k) + \gamma_8\bar\alpha^2\pi_k\|d_k\|^2 \\ &\le -\bar\alpha\Delta m_k(d_k;\pi_k) + \gamma_8\bar\alpha^2\pi_k\|d_k\|^2, \end{aligned} \tag{3.20}$$

and so with (3.19) we have

$$(1-\eta)\Delta m_k(d_k;\pi_k) < \gamma_8\bar\alpha\pi_k\|d_k\|^2. \tag{3.21}$$

From (3.18) we then find

$$(1-\eta)\gamma_7\pi_k\|d_k\|^2 < \gamma_8\bar\alpha\pi_k\|d_k\|^2,$$

which implies $\alpha_k \ge \alpha^S_{\min} \triangleq (1-\eta)\gamma_7/(2\gamma_8)$. Thus, along with Lemma 3.9 and Lemma 3.11, we have

$$\begin{aligned}\tilde\phi(x_{k+1};\pi_{k+1}) &\le \tilde\phi(x_k;\pi_k) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k) \\ &\le \tilde\phi(x_k;\pi_k) - \eta\alpha^S_{min}\gamma_3\|A_k^Tc_k\|^2 \\ &\le \tilde\phi(x_k;\pi_k) - \tfrac{1}{4}\eta\alpha^S_{min}\gamma_3\gamma_6^2. \end{aligned} \tag{3.22}$$

We can now prove the result by showing that there can only be a finite number of iterates with $x_k \in S$. For the purpose of deriving a contradiction, suppose there is an infinite number of iterations with $x_k \in S$. In this case, we have shown that there are an infinite number of iterations with $x_k \in S$ and $k \notin T_2$ during which (3.22) implies a reduction in $\tilde\phi$ by a constant amount. This contradicts the fact that $\tilde\phi$ is bounded below under Assumption 3.1, and so there can only be a finite number of iterates with $x_k \in S$. Therefore, since γ_6 was chosen arbitrarily, this implies (3.9). □

We have the following corollary when this result is combined with Lemma 3.6.

Corollary 3.13. *Algorithm* TRINS *yields*

$$\lim_{k\to\infty}\|v_k\| = 0.$$

We now begin our analysis for particular cases when all limit points of the sequence $\{A_k\}$ produced by Algorithm TRINS have full row rank. That is, we now focus on cases where the singular values of the constraint Jacobians $\{A_k\}$ remain bounded below and away from zero for all k sufficiently large. We prove that the penalty parameter will remain bounded in such situations and that the sequence $\{x_k, \lambda_k\}$ will satisfy the limit (3.8).

We begin this analysis by providing a second corollary to Lemma 3.12.

Corollary 3.14. *Suppose that there exists $k_A \ge 0$ such that the smallest singular values of $\{A_k\}_{k\ge k_A}$ are bounded below and away from zero. Then, Algorithm* TRINS *yields*

$$\lim_{k\to\infty}\|c_k\| = 0. \tag{3.23}$$

Proof. As the singular values of $\{A_k\}_{k\ge k_A}$ are bounded away from zero, there exists $\gamma_9 > 0$ such that

$$\|A_k^Tc_k\| \ge \gamma_9\|c_k\| \tag{3.24}$$

for all $k \geq k_A$. Thus, the result follows from Lemma 3.12. □

We now show that the reductions attained in the linear model of the constraints will eventually remain large in proportion to the norm of the normal component v_k over all iterations when the step satisfies termination test 3. This result is important as the penalty parameter will be increased if and only if this test is satisfied.

LEMMA 3.15. *Suppose that there exists $k_A \geq 0$ such that the smallest singular values of $\{A_k\}_{k \geq k_A}$ are bounded below and away from zero. Then, for all $k \geq k_A$ such that $k \in T_3$, there exists $\gamma_{10} > 0$ such that*

$$\|v_k\| \leq \gamma_{10}(\|c_k\| - \|c_k + A_k d_k\|). \tag{3.25}$$

Proof. As the singular values of $\{A_k\}_{k \geq k_A}$ are bounded away from zero, (3.24) holds for all $k \geq k_A$. Then, from (3.11) and the fact that $\|A_k^T A_k\|$ is bounded above under Assumption 3.1, we have that for some $0 < \gamma_{11} < 1$

$$\begin{aligned} \mathcal{C}_k &= \gamma_1 \frac{\|A_k^T c_k\|^2}{\|c_k\|} \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\} \\ &\geq \gamma_1 \gamma_9^2 \|c_k\| \min\left\{\frac{1}{\|A_k^T A_k\|}, \omega\right\} \\ &\geq \gamma_{11}\|c_k\|. \end{aligned} \tag{3.26}$$

Since v_k lies in the range space of A_k^T, we may express $v_k = A_k^T \hat{v}_k$ satisfying $A_k A_k^T \hat{v}_k = -c_k + q_k$ for some residual vector $q_k \in \mathbb{R}^t$. Since the singular values of $\{A_k\}_{k \geq k_A}$ are bounded away from zero we have for $k \geq k_A$ that

$$\hat{v}_k = (A_k A_k^T)^{-1}(-c_k + q_k).$$

Moreover, from (3.11) and (3.26), it follows that

$$\|q_k\| \leq \|c_k\| - \mathcal{C}_k \leq (1 - \gamma_{11})\|c_k\|.$$

Thus, by (2.16a), (3.11), and (3.26), we find

$$\begin{aligned} \|v_k\| &\leq \|A_k^T\|\|\hat{v}_k\| \\ &\leq \|A_k^T\|\|(A_k A_k^T)^{-1}\|(\|c_k\| + \|q_k\|) \\ &\leq \|A_k^T\|\|(A_k A_k^T)^{-1}\|(2 - \gamma_{11})\|c_k\| \\ &\leq \|A_k^T\|\|(A_k A_k^T)^{-1}\|\tfrac{1}{\gamma_{11}}(2 - \gamma_{11})\mathcal{C}_k \\ &\leq \|A_k^T\|\|(A_k A_k^T)^{-1}\|\tfrac{1}{\gamma_{11}}(2 - \gamma_{11})(\|c_k\| - \|c_k + A_k v_k\|) \\ &\leq \|A_k^T\|\|(A_k A_k^T)^{-1}\|\tfrac{1}{\gamma_{11}\epsilon}(2 - \gamma_{11})(\|c_k\| - \|c_k + A_k d_k\|), \end{aligned}$$

so (3.25) holds for some $\gamma_{10} > 0$ since $\|A_k^T\|$ is bounded under Assumption 3.1 and $\|(A_k A_k^T)^{-1}\|$ is bounded as the singular values of $\{A_k\}_{k \geq k_A}$ are bounded below and away from zero. □

The sequence of penalty parameter values can now be bounded under the conditions of Lemma 3.15.

LEMMA 3.16. *Suppose that there exists $k_A \geq 0$ such that the smallest singular values of $\{A_k\}_{k \geq k_A}$ are bounded below and away from zero. Then, $\pi_k = \bar{\pi}$ for all $k \geq \bar{k}$ for some $\bar{k} \geq k_A$ and $\bar{\pi} < \infty$.*

Proof. We need only consider here iterations where $k \in T_3$ since π_k remains unchanged otherwise. That is, we may assume the inequalities (2.16) and that the tangential component condition (2.10) or (2.11) are satisfied. We show below that there exists $\gamma_{12} > 0$ such that

$$g_k^T d_k + \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} \leq \gamma_{12}\left(\|c_k\| - \|c_k + A_k d_k\|\right), \tag{3.27}$$

and so from (2.17) we have that $\{\pi_k^{trial}\}$ is bounded. This, along with the fact that when Algorithm `TRINS` increases π it does so by at least a positive finite amount, proves the result.

To prove (3.27), we first consider the case where $\|u_k\| \leq \psi\|v_k\|$, so

$$\|d_k\|^2 \leq 2(\|u_k\|^2 + \|v_k\|^2) \leq 2(\psi^2 + 1)\|v_k\|^2.$$

Then, by Lemma 3.15 and since $\|W_k\|$ is bounded under Assumption 3.1, there exist $\gamma_{13}, \gamma_{13}', \gamma_{13}'' > 0$ such that

$$\begin{aligned} g_k^T d_k + \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} &\leq g_k^T d_k + \gamma_{13}\|u_k\|^2 \\ &\leq \|g_k\|\|d_k\| + \gamma_{13}\psi^2\|v_k\|^2 \\ &\leq \|g_k\|\sqrt{2(\psi^2+1)}\|v_k\| + \gamma_{13}\psi^2\|v_k\|^2 \\ &\leq \gamma_{13}'\|v_k\| \\ &\leq \gamma_{13}''(\|c_k\| - \|c_k + A_k d_k\|). \end{aligned}$$

Similarly, if $\|u_k\| > \psi\|v_k\|$, then by Assumption 3.1, Lemmas 3.7 and 3.15, and the tangential component condition (2.11) there exist $\gamma_{14}, \gamma_{14}' > 0$ such that

$$\begin{aligned} g_k^T d_k + \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} &= g_k^T v_k + g_k^T u_k + \tfrac{1}{2}u_k^T W_k u_k \\ &\leq \|g_k\|\|v_k\| - v_k^T W_k u_k + \zeta\|v_k\| \\ &\leq \gamma_{14}\|v_k\| \\ &\leq \gamma_{14}'(\|c_k\| - \|c_k + A_k d_k\|). \end{aligned}$$

Thus, (3.27) follows with $\gamma_{12} = \max\{\gamma_{13}'', \gamma_{14}'\}$. □

We have thus completed our presentation of results specifically for cases when all limit points of $\{A_k\}$ produced by Algorithm `TRINS` have full row rank. In contrast, the next three lemmas simply require that the penalty parameter eventually remains constant, which may occur in situations beyond these.

LEMMA 3.17. *If $\pi_k = \bar{\pi}$ for all $k \geq \bar{k}$ for some $\bar{k} \geq 0$ and $\bar{\pi} < \infty$, then the sequence $\{\alpha_k\}$ is bounded below and away from zero.*

Proof. First, if $d_k = 0$, then $\alpha_k = 1$, so from now on we consider $k \notin T_2$.

As in the proof of Lemma 3.12 (see (3.19)-(3.21)), we have for $k \notin T_2$ that if (2.18) fails for $\bar{\alpha} > 0$, then

$$(1-\eta)\Delta m_k(d_k;\pi_k) < \bar{\alpha}\gamma_8\pi_k\|d_k\|^2.$$

Lemma 3.9 and (3.13) then yield

$$(1-\eta)\gamma_3\left(\|u_k\|^2 + \pi_k\|A_k^T c_k\|^2\right) < \bar{\alpha}\gamma_5\gamma_8\pi_k\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^T c_k\|^2\right),$$

so

$$\bar{\alpha} > \frac{(1-\eta)\gamma_3\left(\|u_k\|^2 + \pi_k\|A_k^T c_k\|^2\right)}{\gamma_5\gamma_8\pi_k\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^T c_k\|^2\right)} \triangleq \alpha_{min},$$

where $\alpha_{min} > 0$ is bounded away from zero for all $k \notin T_2$ as $0 < \pi_{-1} \leq \pi_k \leq \bar{\pi}$. Thus, if $\pi_k = \bar{\pi}$ for all $k \geq \bar{k}$ for some $\bar{k} \geq 0$ and $\bar{\pi} < \infty$, then α_k for $k \notin T_2$ need never be set below $\alpha_{min}/2$ for (2.18) to be satisfied. □

We now show that the primal steps vanish in the limit under the same conditions.

LEMMA 3.18. *If $\pi_k = \bar{\pi}$ for all $k \geq \bar{k}$ for some $\bar{k} \geq 0$ and $\bar{\pi} < \infty$, then*

$$\lim_{k\to\infty} \|d_k\| = 0.$$

Proof. From Lemma 3.17, there exists $\gamma_{15} > 0$ such that

$$\phi(x_k;\pi_k) - \phi(x_k + \alpha_k d_k;\pi_k) \geq \gamma_{15}\Delta m_k(d_k;\pi_k)$$

for all k. Thus, for all $k > \bar{k}$, Lemma 3.9 yields

$$\begin{aligned} \phi(x_{\bar{k}};\bar{\pi}) - \phi(x_k;\bar{\pi}) &= \sum_{j=\bar{k}}^{k-1} (\phi(x_j;\bar{\pi}) - \phi(x_{j+1};\bar{\pi})) \\ &\geq \gamma_{15} \sum_{j=\bar{k}}^{k-1} \Delta m_j(d_j;\bar{\pi}) \\ &\geq \gamma_{15}\gamma_3 \sum_{j=\bar{k}, j\notin T_2}^{k-1} \left(\|u_j\|^2 + \bar{\pi}\|A_j^T c_j\|^2\right) \end{aligned}$$

Thus, since $u_k = 0$ for $k \in T_2$ and $\phi(x;\bar{\pi})$ is bounded below under Assumption 3.1, we have

$$\lim_{k\to\infty} \|u_k\| = 0.$$

The result follows from this limit and Corollary 3.13. □

The above results now allow us to show that in cases where the penalty parameter eventually remains constant, dual feasibility is attained in the limit.

LEMMA 3.19. *If $\pi_k = \bar{\pi}$ for all $k \geq \bar{k}$ for some $\bar{k} \geq 0$ and $\bar{\pi} < \infty$, then*

$$\lim_{k\to\infty} \|g_k + A_k^T \lambda_{k+1}\| = 0.$$

Proof. From (2.19) and the first block equation of (2.9), we find with $\kappa' = \max\{\kappa_a, \kappa_b\} \in (0,1)$ that from Assumption 3.1, (2.13), (2.15), and (2.16) that there exist $\gamma_{16}, \gamma_{16}' > 0$ such that

$$\begin{aligned} \|g_k + A_k^T\lambda_{k+1}\| &\leq \|g_k + A_k^T(\lambda_k + \delta_k)\| \\ &= \|\rho_k - W_k d_k\| \\ &\leq \|\rho_k\| + \gamma_{16}\|d_k\| \\ &\leq \max\left\{\kappa' \left\| \begin{bmatrix} g_{k-1} + A_{k-1}^T\lambda_k \\ A_{k-1}v_{k-1} \end{bmatrix} \right\|, \beta(\|c_k\| - \|c_k + A_k v_k\|)\right\} + \gamma_{16}\|d_k\| \\ &\leq \kappa'\|g_{k-1} + A_{k-1}^T\lambda_k\| + \gamma_{16}' \max\{\|v_{k-1}\|, \|v_k\|, \|d_k\|\}. \end{aligned}$$

Consider an arbitrary $\gamma_{17} > 0$. Corollary 3.13 and Lemma 3.18 imply that there exists $k' > 0$ such that for all $k \geq k'$ we have $\gamma_{16}' \max\{\|v_{k-1}\|, \|v_k\|, \|d_k\|\} \leq (1-\kappa')\gamma_{17}/2$.

Suppose $k \geq k'$ and $\|g_{k-1} + A_{k-1}^T \lambda_k\| > \gamma_{17}$. Then the above yields

$$\begin{aligned} \|g_k + A_k^T \lambda_{k+1}\| \leq\ & \kappa' \|g_{k-1} + A_{k-1}^T \lambda_k\| + \tfrac{1-\kappa'}{2}\gamma_{17} \\ =\ & \|g_{k-1} + A_{k-1}^T \lambda_k\| - (1-\kappa')\|g_{k-1} + A_{k-1}^T \lambda_k\| + \tfrac{1-\kappa'}{2}\gamma_{17} \\ <\ & \|g_{k-1} + A_{k-1}^T \lambda_k\| - \tfrac{1-\kappa'}{2}\gamma_{17}. \end{aligned}$$

Therefore, $\{\|g_k + A_k^T \lambda_{k+1}\|\}$ decreases monotonically by at least a constant amount for $k \geq k'$ while $\{\|g_k + A_k^T \lambda_{k+1}\|\} > \gamma_{17}$, so we eventually find $\|g_k + A_k^T \lambda_{k+1}\| \leq \gamma_{17}$ for some $k = k'' > k'$. Then, for $k \geq k''$ we find

$$\|g_k + A_k^T \lambda_{k+1}\| \leq\ \kappa' \gamma_{17} + \tfrac{1-\kappa'}{2}\gamma_{17} \leq \gamma_{17}$$

and so by induction $\|g_k + A_k^T \lambda_{k+1}\| \leq \gamma_{17}$ for all $k \geq k''$. Since $\gamma_{17} > 0$ was chosen arbitrarily, the result follows. □

We are now ready to prove the main result stated at the beginning of this section.

Proof. (Theorem 3.3) If all limit points of $\{A_k\}$ have full row rank, then there exists $k_A \geq 0$ such that the smallest singular values of $\{A_k\}_{k \geq k_A}$ are bounded below and away from zero. By Lemma 3.16 we then have that the penalty parameter is constant for all k sufficiently large, which means $\{\pi_k\}$ is bounded, and so by Corollary 3.14 and Lemma 3.19 we have the limit (3.8).

If a limit point of $\{A_k\}$ does not have full row rank, then we have the limit (3.9) by Lemma 3.12. Moreover, if $\{\pi_k\}$ is bounded, then the fact if Algorithm TRINS increases π_k is does so by at least a positive finite amount implies that the penalty parameter is in fact constant for all k sufficiently large, which implies with Lemma 3.19 that we have the limit (3.10). □

4. An Implementation.

Algorithm TRINS was implemented in Matlab for the purposes of numerical experimentation. In this section we describe the particular aspects of our code.

First, we present appropriate stopping conditions for the `for` loop of the algorithm. We terminate if one of the following conditions is satisfied:

$$\frac{\|g_k + A_k^T \lambda_k\|_\infty}{\max\{\|g_0\|_\infty, 1\}} \leq \epsilon_{dual} \quad \text{and} \quad \frac{\|c_k\|_\infty}{\max\{\|c_0\|_\infty, 1\}} \leq \epsilon_{prim}, \tag{4.1a}$$

$$\frac{\|g_k + A_k^T \lambda_k\|_\infty}{\max\{\|g_0\|_\infty, 1\}} \leq \epsilon_{dual} \quad \text{and} \quad \frac{\|A_k^T c_k\|_\infty}{\max\{\|A_k\|_\infty \|c_k\|_\infty, 1\}} \leq \epsilon_{inf,1}, \tag{4.1b}$$

$$\pi > \pi_{max} \quad \text{and} \quad \frac{\|A_k^T c_k\|_\infty}{\max\{\|A_k\|_\infty \|c_k\|_\infty, 1\}} \leq \epsilon_{inf,2}, \tag{4.1c}$$

$$\text{or} \quad k > k_{max}, \tag{4.1d}$$

where

$$\begin{aligned} &\epsilon_{dual} \leftarrow 10^{-6},\ \epsilon_{prim} \leftarrow 10^{-6},\ \epsilon_{inf,1} \leftarrow 10^{-7}, \\ &\epsilon_{inf,2} \leftarrow 10^{-4},\ \pi_{max} \leftarrow 10^{10},\ \text{and } k_{max} \leftarrow 10^3. \end{aligned}$$

If condition (4.1a) is satisfied, then we return that a feasible and optimal solution has been found. Otherwise, if (4.1b) or (4.1c) is satisfied, then the problem is deemed locally infeasible. We use $\epsilon_{inf,2} > \epsilon_{inf,1}$ as the algorithm may become numerically unstable as π becomes very large. Finally, if (4.1d) is satisfied, then we terminate due to a limit on the number of outer iterations.

Notice that the conditions in (4.1) may return a "false infeasible", i.e., a message that the problem is locally infeasible when further iterations may produce a sufficiently feasible iterate. In practice one may, for example, utilize a smaller value for $\epsilon_{inf,1}$ and/or a larger value for π_{max} to ensure that such a result is not returned, but in our tests the ratio $\epsilon_{prim}/\epsilon_{inf,1} = 10$ is justified by the fact that prior to the employment of our approach we prescale the objective and constraint functions so that the first derivative of each has a ℓ_∞-norm less than or equal to 10; see [26]. As a result, no "false" declaration of infeasibility was reported in the tests described below.

In terms of the input parameters required in our approach, we direct the reader to [3, 4] for general guidelines for the selection of these values. In particular, the inexactness parameters κ_a, κ_b, ϵ_b, ϵ, and β are vitally important and should ideally be tuned for each application. In our experiments here, they are simply set to default values along with the parameters τ, η, and δ_π. The parameter ψ should generally be set to a positive value less than one as this quantity determines the maximum value of $\|u_k\|/\|v_k\|$ for which we consider the complete primal step d_k to be sufficiently *normal.* Here we set this value to 0.1. In our experiments we found the parameter ζ to be inconsequential for our given values of the remaining parameters, so here we pick $\zeta \leftarrow 0.1$.

The parameter ω should be chosen to reflect the conditioning of the constraint Jacobian during each iteration. Thus, one may set it as a constant, or one may consider setting the value dynamically using traditional ideas from trust region methods, with the added restriction that the value does not stray from a prescribed interval. To emulate the latter approach in our code, we initialize $\omega_0 \leftarrow 10^3$ and update this value at the end of iteration k according to the rule

$$\omega_{k+1} \leftarrow \begin{cases} \min\{2\omega_k, 10^{20}\} & \text{if } \|v_k\| = \omega_k \|A_k^T c_k\| \text{ and } \alpha_k = 1 \\ \omega_k & \text{otherwise.} \end{cases}$$

Finally, we set $\theta \leftarrow 10^{-12}$ as a relatively low tolerance for the curvature condition (2.11a) to avoid unnecessary modifications of the Hessian close to a solution that might interfere with fast local convergence.

A complete listing of all the parameters used in our code is given in Table 4.1.

Parameter	Value	Parameter	Value
κ_a	10^{-4}	θ	10^{-12}
ϵ	$1 - 10^{-4}$	ζ	10^{-1}
τ	10^{-1}	β	10
κ_b	10^{-2}	ϵ_b	10^{-2}
η	10^{-8}	δ_π	10^{-4}
ω_0	10^3	σ	$\tau\epsilon$
ψ	10^{-1}	π_{-1}	10^{-1}

Table 4.1
Parameter values used for Algorithm `TRINS`

As mentioned in §2, an efficient method for the normal step computation, i.e., a method for the approximate solution of problem (2.2), is the conjugate gradient (CG) algorithm with Steihaug stop tests [24]. A direct implementation of this approach, however, can become numerically unstable if A_k is ill-conditioned or rank deficient, and so the normal step in our implementation was performed with an adapted version of the `LSQR` algorithm of Paige and Saunders [18]. This approach is mathematically equivalent to CG, but has better numerical properties when applied to (near) singular systems. We extended the implementation in [19] so that the iteration was terminated if a direction of zero curvature was found (at which point the current LSQR iterate was returned), the trust region constraint was violated (at which point the point at which the boundary was crossed was returned as the solution), or a solution v_k satisfying $\|c_k + A_k v_k\| \leq 10^{-4}\|c_k\|$ was computed.

The primal-dual step computation of Algorithm TRINS was performed with an adapted version of the implementation in [20] of the Minimum Residual (MINRES) method [17]. For most runs of Algorithm TRINS performed in our experiments, this code was sufficient for computing acceptable search directions (i.e., steps satisfying termination test 1, 2, or 3) despite the fact that a preconditioner was not implemented in our code. Occasionally, however, unpreconditioned MINRES was not able to provide a step satisfying one of our termination criteria before the limit of $2(n+t)$ iterations was reached, so in such cases we employed an adapted version of Kelley's implementation [14] of the Generalized Minimum Residual (GMRES) method [23] to attempt to compute an acceptable step. This latter approach is not used initially as it does not take advantage of the symmetry of the primal-dual equations (2.6), but we have found it to be more numerically stable than MINRES on (near) singular systems.

A few comments are necessary to describe our method for perturbing W_k within a run of the primal-dual step computation. We begin the process by setting W_k to the exact Hessian of the Lagrangian and the initial guess set as the zero vector. If this initial vector does not satisfy a termination test, then MINRES iterations are performed until either an acceptable step is found or a modification to W_k is deemed appropriate. If a modification to W_k is made, we restart the MINRES solver with the solution initialized to the last solution computed for the previous W_k. Overall, for any W_k we perform at most $2(n+t)$ MINRES iterations. In order to avoid having our modification strategy perturb the matrix prematurely, we only allow a modification to the current W_k once (2.13) is satisfied with κ set to $1/2$, or once $(n+t)/2$ iterations have been performed. Finally, as for the specific form of the modification, we set $W_k \leftarrow W_k + \mu I$ with μ set according to the rules outlined in [26].

The primal-dual step computation is summarized in the following algorithm, which covers the repeat loop in the description given in §2. Note that for termination test 1 we use the norm of the entire primal-dual residual (ρ_k, r_k) on the left-hand-side of (2.13) with the hope that this choice will yield fast local convergence.

```
ALGORITHM TRINS-step: STEP COMPUTATION FOR ALGORITHM TRINS
Set $j \leftarrow 0$, $(d^0, \delta^0) \leftarrow (v_k, 0)$, $W_k \leftarrow \nabla^2_{xx}\mathcal{L}_k$, and choose $\mu > 0$
if termination test 1, 2, or 3 is satisfied, return $(d_k, \delta_k) \leftarrow (d^j, \delta^j)$
while $j < 2(n+t)$
    Increment $j \leftarrow j + 1$
    Perform a MINRES iteration on (2.6) to compute $(d^j, \delta^j)$
    if $j \geq (n+t)/2$ and (2.13) holds for $\kappa = 1/2$
        if termination test 1 or 3 is satisfied, then break
        if termination test 2 is satisfied, then set $d^j \leftarrow 0$ and break
        if $(d^j, \delta^j)$ does not satisfy (2.10) or (2.11a)
          Set $j \leftarrow 0$, $(d^0, \delta^0) \leftarrow (d^j, \delta^j)$, $W_k \leftarrow W_k + \mu I$, and choose $\mu > 0$
        endif
    endif
endwhile
return $(d_k, \delta_k) \leftarrow (d^j, \delta^j)$
```

The algorithm does not change when GMRES replaces MINRES as the iterative solver.

4.1. Numerical Experiments. Algorithm TRINS is designed for very large applications, but its effectiveness can be illustrated on problems of any size. Thus, we chose to test our Matlab implementation on a variety of problems in the CUTEr [1, 12] collection. From this set, we selected all of the equality constrained problems for which AMPL [11] models were available that have at least one degree of freedom. This latter requirement was enforced to allow us to create a complete set of perturbed models, as described below, but we note that the 2 problems available that did not satisfy this condition (bt10 and hs008) were solved with no issue by our code. The selected set is composed of 73 problems. We found that all problems in this set have sufficiently feasible and optimal solutions according to (4.1a), so the goal here is to illustrate the robustness of the approach without a fine-tuning of the parameters and algorithmic components for each problem instance.

We performed three related sets of numerical experiments to illustrate the versatility of Algorithm TRINS. In the first we applied our implementation of Algorithm TRINS with the fixed set of input parameters given in Table 4.1 to the problems just described. The second set of experiments was performed with the same set of problems and the same inputs, but with a perturbation to the constraint functions in the problem formulations. In particular, in each model we split the first constraint $c_1(x) = 0$ into the pair of constraints

$$c_1(x) = 0 \quad \text{and} \tag{4.2a}$$

$$c_1(x) - c_1^2(x) = 0. \tag{4.2b}$$

Each problem in this set therefore also has a sufficiently feasible and optimal solution, but the constraint Jacobians will be rank deficient everywhere and the linearized constraint functions will be inconsistent at all points with $c_1(x) \neq 0$. The third set of experiments is similar to the second set, except that the right-hand-side of the added constraint (4.2b) is set to 1, thus creating an infeasible instance of each model.

We compare the results of Algorithm TRINS to the results of the method described in [3], which has no features for dealing with Jacobian singularity, on all three sets of problems to illustrate that Algorithm TRINS can both perform well on well-conditioned problems and overcome the obstacles posed by rank deficient constraint formulations.

Problem data and results for each run are given in Tables 4.2 and 4.3. Here, n represents the number of variables and t represents the number of constraints; i.e., the number of constraints in each perturbed constraint set is $t + 1$. We provide both a termination message and outer iteration counts for Algorithm TRINS and the algorithm described in [3], which we refer to as Algorithm INS. The Original columns correspond to results for the original models, the Perturbed columns corresponds to results for the second set of experiments, and the Infeasible column corresponds to results for the third set of experiments. We do not provide an Infeasible column for Algorithm INS as the algorithm was not able to satisfy (4.1a), (4.1b), or (4.1c) for more than a few problem instances. This is not surprising, however, as that method will stall when feasibility is not attained. Notice that we would ideally have the algorithms return that an optimal solution was found, denoted by opt, in all instances in columns Original and Perturbed, and that we would ideally have Algorithm TRINS return that (4.1b) or (4.1c) was satisfied, denoted by inf and π, respectively, in all instances in column Infeasible. In a few instances, however, the results are not ideal, and occasionally we find that the iteration limit (4.1d) was reached, which we denote by ---.

			Algorithm TRINS						Algorithm INS			
			Original		Perturbed		Infeasible		Original		Perturbed	
Name	n	t	Res	Itr	Res	Itr	Res	Itr	Res	Itr	Res	Itr
aug2d	20192	9996	opt	28	opt	6	π	12	opt	6	opt	6
aug3dc	3873	1000	opt	4	opt	5	π	9	opt	4	opt	6
aug3d	3873	1000	opt	4	opt	5	π	8	opt	4	opt	8
bt11	5	3	opt	8	opt	9	π	7	opt	8	---	---
bt12	5	3	opt	3	opt	7	π	13	opt	4	---	---
bt1	2	1	opt	50	opt	25	inf	58	---	---	---	---
bt2	3	1	opt	11	opt	12	opt	12	opt	11	---	---
bt3	5	3	opt	2	opt	6	π	6	opt	2	---	---
bt4	3	2	opt	14	opt	44	---	---	opt	10	---	---
bt5	3	2	opt	7	opt	9	π	9	opt	7	---	---
bt6	5	2	opt	9	opt	17	π	17	opt	9	---	---
bt7	5	3	opt	32	opt	34	π	48	opt	21	---	---
bt8	5	2	opt	10	opt	7	inf	98	opt	10	---	---
bt9	4	2	opt	23	opt	156	π	25	opt	67	---	---
byrdsphr	3	2	opt	19	---	---	π	24	opt	30	---	---
catena	32	11	opt	18	---	---	---	---	opt	31	---	---
dixchlng	10	5	opt	10	opt	9	inf	23	opt	10	---	---
dtoc1l	14985	9990	opt	8	opt	8	π	9	opt	8	opt	8
dtoc1na	1485	990	opt	7	opt	7	π	7	opt	7	opt	7
dtoc1nb	1485	990	opt	6	opt	6	π	9	opt	6	opt	6
dtoc1nc	1485	990	opt	12	opt	12	π	10	opt	9	opt	9
dtoc1nd	735	490	opt	61	opt	44	π	18	---	---	---	---
dtoc2	5994	3996	opt	37	opt	29	π	14	opt	16	opt	31
dtoc3	14996	9997	opt	20	opt	12	inf	26	opt	3		
dtoc4	14996	9997	opt	32	opt	32	inf	15	opt	3		
dtoc5	9998	4999	opt	21	opt	10	inf	22	opt	4		
dtoc6	10000	5000	opt	63	opt	17	π	13	opt	20		
eigena2	110	55	opt	23	opt	24	π	42	opt	18	opt	19
eigenaco	110	55	opt	4	opt	4	inf	16	opt	4	opt	4
eigenb2	110	55	opt	13	opt	17	π	528	opt	28	opt	15
eigenbco	110	55	opt	521	---	---	π	152	---	---	opt	65
eigenc2	462	231	opt	18	opt	23	π	21	opt	20	---	---
eigencco	30	15	opt	14	opt	16	π	59	opt	16	---	---
fccu	19	8	opt	3	opt	5	inf	12	opt	3	---	---
genhs28	10	8	opt	3	opt	3	inf	13	opt	3	opt	3
gilbert	1000	1	opt	38	opt	20	opt	20	opt	20	---	---

Table 4.2

Termination results for Algorithm TRINS *and Algorithm* INS *(described in [3]) for problems from the CUTEr collection. The original problems are solved along with related models where the constraint set has been perturbed.*

The results suggest that Algorithm TRINS is versatile and robust and can successfully be applied to nonconvex problems with (near) rank deficient constraint Jacobians. Moreover, the differences between the success rates for the two methods illustrate the need for the algorithmic extensions described in this work. Algorithm TRINS failed to satisfy (4.1a), (4.1b), or (4.1c) before the iteration limit was reached for a few problems instances, but we conjecture that many of these unsuccessful runs could be remedied by a more sophisticated implementation.

5. Conclusion. In this paper we have proposed and analyzed a matrix-free primal-dual algorithm for large-scale equality constrained optimization. The method extends the application of the sufficient merit function approximation reduction termination (SMART) Tests developed in [3, 4] to problems where the constraint Jacobians may lose rank during the solution process. A crucial aspect of the approach is the

			Algorithm TRINS						Algorithm INS			
			Original		Perturbed		Infeasible		Original		Perturbed	
Name	n	t	Res	Itr	Res	Itr	Res	Itr	Res	Itr	Res	Itr
gridnetb	13284	6724	opt	10	opt	8	π	10	opt	4		
hager1	10000	5000	opt	4	opt	7	---	---	opt	3		
hager2	10000	5000	opt	2	opt	4	opt	4	opt	2		
hager3	10000	5000	opt	2	opt	5	opt	5	opt	2		
hs006	2	1	opt	2	opt	21	π	20	opt	2	---	---
hs007	2	1	opt	23	opt	21	inf	25	opt	287	---	---
hs009	2	1	opt	3	opt	3	inf	11	opt	4	opt	4
hs026	3	1	opt	14	opt	32	π	17	opt	14	---	---
hs027	3	1	opt	39	opt	566	π	14	opt	79	---	---
hs028	3	1	opt	1	opt	1	inf	10	opt	1	opt	1
hs039	4	2	opt	23	opt	156	π	25	opt	67	---	---
hs040	4	3	opt	4	opt	4	π	8	opt	4	---	---
hs046	5	2	opt	14	opt	17	π	21	opt	14	---	---
hs047	5	3	opt	16	opt	16	---	---	---	---	---	---
hs048	5	2	opt	1	opt	1	inf	10	opt	1	opt	1
hs049	5	2	opt	12	opt	12	π	9	opt	12	opt	12
hs050	5	3	opt	8	opt	8	π	9	opt	8	opt	8
hs051	5	3	opt	2	opt	2	inf	11	opt	2	opt	2
hs052	5	3	opt	2	opt	6	π	7	opt	2	---	---
hs061	3	2	opt	8	opt	10	π	10	opt	29	---	---
hs077	5	2	opt	9	opt	36	π	77	opt	9	---	---
hs078	5	3	opt	5	opt	6	π	263	opt	4	opt	244
hs079	5	3	opt	4	opt	7	π	5	opt	4	---	---
hs100lnp	7	2	opt	7	opt	7	inf	8	opt	8	---	---
hs111lnp	10	3	opt	135			π	28	opt	766	---	---
lch	600	1	opt	27	opt	57	---	---	opt	29	opt	6
maratos	2	1	opt	5	opt	7	π	8	opt	5	---	---
mwright	5	3	opt	7	opt	7	π	11	opt	8	---	---
orthrdm2	4003	2000	opt	12	opt	6	inf	7	opt	7	opt	7
orthrds2	203	100	opt	119	opt	53	π	42	opt	78	---	---
orthrega	517	256	opt	70	opt	55	π	81	opt	51	opt	50
orthregb	27	6	opt	2	opt	4	inf	6	opt	2	---	---
orthregc	10005	5000	opt	205	opt	16	π	11	opt	15		
orthregd	10003	5000	opt	20	opt	7	π	7	opt	9		
orthrgdm	10003	5000	opt	28	opt	8	π	8	opt	10		
orthrgds	10003	5000	opt	73	---	---	---	---	opt	32		
robot	14	9	opt	40	---	---	π	14	opt	6	opt	10

TABLE 4.3

Termination results for Algorithm TRINS *and Algorithm* INS *(described in [3]) for problems from the CUTEr collection. The original problems are solved along with related models where the constraint set has been perturbed.*

introduction of a trust region subproblem for the calculation of a normal step component, after which the (approximate) solution of a carefully constructed primal-dual system can be used to make the algorithm well-defined and globally convergent to first-order optimal points, or at least to stationary points of a feasibility measure. We close by discussing further some practical issues related to the implementation of our approach.

A drawback of the algorithm described in this paper is that the trust region constraint may block the approach from nearly satisfying $c_k + A_k d_k = 0$ in cases when doing so is possible. This phenomenon may obstruct the algorithm from acting like an inexact Newton approach, and so may hinder the possibility for fast local convergence to a solution point. Thus, a more efficient implementation may first attempt to compute a step via the Newton system (1.4) using the termination criteria described in

[3], and only revert to the approach described in this paper if the calculations indicate that the linear model of the constraints may be ill-conditioned or inconsistent.

A further enhancement that may improve the practical performance of our approach is the inclusion of a watchdog strategy to avoid the Maratos effect. Moreover, our and other implementations would benefit from effective preconditioners for the iterative solution of (2.2) and (2.6), and an alternative to the iterative linear solvers `MINRES` and `GMRES` is the recently developed `MINRESQLP` method, which is a more numerically stable version of the `MINRES` algorithm described by Choi in [8].

REFERENCES

[1] I. Bongartz, A. R. Conn, N. Gould, and Ph.L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21(1):123–160, 1995.

[2] J. V. Burke. A robust trust region method for constrained nonlinear programming problems. *SIAM Journal on Optimization*, 2(2):324–347, 1992.

[3] R. H. Byrd, F. E. Curtis, and J. Nocedal. An inexact newton method for nonconvex equality constrained optimization. *Mathematical Programming, Series A*, 2008. (to appear).

[4] R. H. Byrd, F. E. Curtis, and J. Nocedal. An inexact SQP method for equality constrained optimization. *SIAM Journal on Optimization*, 19(1):351–369, 2008.

[5] R. H. Byrd, J.-Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[6] R. H. Byrd, J. Nocedal, and R. A. Waltz. Steering penalty methods. *Optimization Methods and Software*, 23(2):197–213, 2008.

[7] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984*, pages 71–82, Philadelphia, PA, USA, 1985. SIAM.

[8] S.-C. Choi. *Iterative Methods for Singular Linear Equations and Least-Squares Problems.* PhD thesis, Stanford University, Palo Alto, CA, USA, 2006.

[9] J.-P. Dussault. Numerical stability and efficiency of penalty algorithms. *SIAM Journal on Numerical Analysis*, 32(1):296–317, 1995.

[10] R. Fletcher. *Practical Methods of Optimization.* J. Wiley and Sons, Chichester, England, Second edition, 1987.

[11] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Brooks/Cole, Second edition, 2002.

[12] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[13] M. Heinkenschloss and D. Ridzal. An inexact trust-region SQP methods with applications to PDE-constrained optimization. In O. Steinbach and G. Of, editors, *Numerical Mathematics and Advanced Applications: Proceedings of Enumath 2007, the 7th European Conference on Numerical Mathematics and Advanced Applications, Graz, Austria*, Heidelberg, Germany, 2007. Springer-Verlag. (submitted).

[14] C. T. Kelley. Iterative methods for linear and nonlinear equations: Matlab codes, 1997.

[15] J. Nocedal and S. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, Second edition, 2006.

[16] E. O. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints.* PhD thesis, University of Colorado at Boulder, Boulder, CO, USA, 1989.

[17] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[18] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[19] C. C. Paige and M. A. Saunders. LSQR: Sparse equations and least squares, 2002.

[20] C. C. Paige and M. A. Saunders. MINRES: Sparse symmetric equations, 2004.

[21] M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49(2):189–213, 1990.

[22] D. Ridzal. *Trust Region SQP Methods with Inexact Linear System Solves for Large-Scale Optimization.* PhD thesis, Rice University, Houston, TX, USA, 2006.

[23] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving

nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[24] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[25] S. Ulbrich. Generalized SQP-methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time PDE-Constrained Optimization*, Philadelphia, PA, USA, 2008. SIAM. (to appear).

[26] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, Series A and B*, 106(1):25–57, 2006.

[27] H. Yamashita and T. Tanabe. A primal-dual exterior point method for nonlinear optimization. *SIAM Journal on Optimization*, 2007. (submitted for publication).