SIAM J. OPTIM.
Vol. 20, No. 3, pp. 1224–1249

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 constrained optimization that is robust and efficient even for problems with (nearly) rank-deficient Jacobian matrices. The method is matrix-free (i.e., it does not require explicit storage or factorizations of derivative matrices), allows for inexact step computations, and is applicable for 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. We show that the algorithm is globally convergent to first-order optimal points or to stationary points of an infeasibility 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

DOI. 10.1137/08072471X

1. Introduction.

A variety of sophisticated algorithms exist for the solution of large-scale nonlinear optimization problems. Some of these algorithms are equipped to cope with (nearly) rank-deficient Jacobian matrices. In some cases this is done by regularizing the constraints with a penalty function [7, 12, 13, 29], and in others it is accomplished by employing trust region techniques [3, 8, 18, 23]. A major drawback of most of these approaches is that they require explicit storage and factorizations of large iteration matrices throughout the solution process, which makes their use impractical for certain classes of problems.

In this paper, we design and analyze an efficient matrix-free algorithm for equality constrained problems of the form

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

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are sufficiently smooth functions. We categorize an optimization algorithm as matrix-free if the approach does not require the explicit storage or factorization of any matrix. Fortunately, as can be seen in the algorithms proposed in [4, 5, 16, 24, 27], such computationally expensive tasks can be avoided by employing iterative solution procedures for the step computations in 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 those associated with the application of a preconditioner and to matrix-vector products with

*Received by the editors May 20, 2008; accepted for publication (in revised form) June 18, 2009; published electronically September 16, 2009.
http://www.siam.org/journals/siopt/20-3/72471.html
†Courant Institute of Mathematical Sciences, New York University, New York, NY 10003 (fecurt@gmail.com). This author was supported by National Science Foundation grant DMS 0602235.
‡Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 (nocedal@eecs.northwestern.edu). 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 10598 (andreasw@watson.ibm.com).

the Jacobian of the constraint functions, its transpose, and the (approximate) Hessian of the Lagrangian function

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

Our focus is on 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. The algorithm proposed in this paper has 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 stationary points of the infeasibility measure

$$\varphi(x) \triangleq \|c(x)\|_2. \tag{1.2}$$

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

$$\nabla\mathcal{L}(x,\lambda) = \begin{bmatrix} g(x) + A(x)^T\lambda \\ c(x) \end{bmatrix} = 0 \tag{1.3}$$

for some vector of Lagrange multipliers $\lambda \in \mathbb{R}^t$. An infeasible stationary point of (1.1) can be classified as one satisfying

$$\|c(x)\|_2 > 0 \quad \text{and} \quad \nabla\varphi(x) = \frac{1}{\|c(x)\|_2} A(x)^T c(x) = 0. \tag{1.4}$$

We would like our method to emulate the fast local convergence behavior of Newton's method applied to the optimality conditions (1.3). With $c^i(x)$ and λ^i denoting the ith constraint function and its corresponding dual variable, respectively, we define

$$W(x,\lambda) \triangleq \nabla^2_{xx}\mathcal{L}(x,\lambda) = \nabla^2_{xx} f(x) + \sum_{i=1}^{t} \lambda^i \nabla^2_{xx} c^i(x) \tag{1.5}$$

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 [17]):

$$\begin{bmatrix} W(x_k,\lambda_k) & A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g(x_k) + A(x_k)^T\lambda_k \\ c(x_k) \end{bmatrix}. \tag{1.6}$$

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

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \ & f(x_k) + g(x_k)^T d + \tfrac{1}{2} d^T W(x_k,\lambda_k) d \\ \text{s.t.} \ & c(x_k) + A(x_k) d = 0. \end{aligned} \tag{1.7}$$

Unfortunately, when $W(x_k,\lambda_k)$ and $A(x_k)$ are not stored explicitly, an exact solution to (1.6) 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 (1.6) as those of (1.7). 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 [5]. The defining characteristic of these approaches is that the primal-dual step computation is performed with an iterative procedure for solving the primal-dual equations (1.6), 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 dynamically during the solution process. The model of ϕ about x_k has the form

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

and the reductions obtained in m_k can be computed easily for any d_k as

$$\begin{aligned} \Delta m_k(d_k;\pi) &\triangleq m_k(0;\pi) - m_k(d_k;\pi) \\ &= -g(x_k)^T d_k + \pi(\|c(x_k)\|_2 - \|c(x_k) + A(x_k)d_k\|_2). \end{aligned} \tag{1.8}$$

If an approximate solution to (1.6) yields a sufficiently large value for $\Delta m_k(d_k;\pi_k)$ for an appropriate value of π_k, then this solution 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 [4, 5], is that we do not assume that the constraint Jacobians $\{A_k\}$ have full row rank throughout the solution process. We also want to guarantee that the algorithm remains well-defined when applied to an instance of problem (1.1) that is locally infeasible. The approach we propose 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 (1.6) since this system may be ill-conditioned or even inconsistent during a particular iteration. The basis of our technique is to include a trust region subproblem for computing a reference step toward satisfying the linearized constraints in (1.7), after which a perturbed primal-dual system is solved (approximately) to produce the search direction. The result is that the step computation is effectively regularized and convergence toward a solution or infeasible stationary point of (1.1) can be guaranteed.

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

Notation. All norms are considered 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 our approach by briefly summarizing the methodology developed in [5] for nonconvex optimization. The central tenet of this approach is that a sufficiently accurate solution (d_k, δ_k) to the primal-dual equations (1.6) 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 $\theta > 0$ and $\sigma \in (0,1)$ and an appropriate π_k. (We have simplified (2.1) slightly from [5] for ease of exposition.) We also monitor the following norm of a linear model of the constraints:

$$l_k(d) \triangleq \|c_k + A_k d\|.$$

If W_k is sufficiently positive definite and a reduction in l_k is obtained, then (2.1) simplifies (the first terms in the max expressions dominate) and requires only that the reduction in m_k is sufficiently large with respect to $\|c_k\|$ and a quadratic term corresponding to the objective of (1.7). However, since W_k may not be positive definite and an inexact solution may not yield a reduction in l_k, 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, which can have the effect of increasing the latter term of $\Delta m_k(d_k;\pi_k)$ as written in (1.8), but we allow this only under certain conditions.

If the singular values of $\{A_k\}$ are bounded away from zero over all k, then the procedure in [5] provides 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. 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 to 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 from an adapted system of primal-dual equations. 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 (1.6) and of the SQP subproblem (1.7) in convex regions. Otherwise, the safeguards embedded in the step computation will at least provide a sufficient reduction in m_k. Appropriate values for all of the input parameters defined in the discussion below are provided along with the details of our implementation in section 4.

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

$$\min_{v\in\mathbb{R}^n} \ \tfrac{1}{2}\|c_k + A_k v\|^2 \quad \text{s.t. } \|v\| \le \omega\|A_k^T c_k\| \tag{2.2}$$

for a given constant $\omega > 0$. The trust region constraint in this problem is not of the traditional type, but we prefer this form to simplify the analysis (note its connection to the stationarity condition (1.4)). Indeed, similar approaches have been followed by other authors; e.g., see [26] and the references therein. An efficient method for computing an approximate solution to problem (2.2) in our context of matrix-free optimization is the conjugate gradient (CG) method with Steihaug stopping tests; see section 4 and [25]. In general, however, we simply assume that v_k satisfies the following condition.

NORMAL COMPONENT CONDITION. *The component v_k must be feasible for problem* (2.2) *and satisfy the Cauchy decrease condition*

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

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

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

Since $\alpha^c = 0$ is feasible for (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, we modify the primal-dual system (1.6) to be

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

Here, W_k is the Hessian of the Lagrangian (1.5) at (x_k, λ_k), or is an approximation to it, so that we may assume $\{W_k\}$ is bounded over all k (see Assumption 3.1 below). We define the tangential component of the step as

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

For an exact solution of (2.6), this tangential component will lie in the null space of A_k (i.e., $A_k u_k = 0$) and the reduction obtained by d_k in l_k 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 (1.7); namely, d_k and u_k solve

$$\left\{ \begin{aligned} &\min_{d \in \mathbb{R}^n} g_k^T d + \tfrac{1}{2} d^T W_k d \\ &\text{s.t. } A_k d = A_k v_k \end{aligned} \right\} \quad \text{and} \quad \left\{ \begin{aligned} &\min_{u \in \mathbb{R}^n} (g_k + W_k v_k)^T u + \tfrac{1}{2} u^T W_k u \\ &\text{s.t. } A_k u = 0 \end{aligned} \right\}, \tag{2.8}$$

respectively.

In our algorithm, we employ an iterative procedure to solve (2.6). Each iteration of this procedure computes an approximate solution (d_k, δ_k) yielding

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

where (ρ_k, r_k) denotes the residual vector. 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 decide when a given inexact solution is an acceptable search direction.

Our conditions for these purposes take the form of three termination tests for an iterative solver applied to (2.6). Once one of these three tests is satisfied, the step computation may be terminated as an acceptable search direction has been found. Before stating these tests, however, we motivate and describe three conditions that will be incorporated into them.

Our first such condition outlines some necessary characteristics of the tangential component u_k of the primal step d_k.

Tangential component condition. *A component* u_k *must satisfy either*

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

or

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

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

where $\psi \geq 0$, $\theta > 0$, *and* $\zeta \geq 0$ *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 the tangential component u_k is sufficiently small in length compared to the normal component v_k. Alternatively, the tangential component may be acceptable provided it yields a low enough objective value to problem (2.8) (see (2.11b)) and the curvature is sufficiently positive along this direction (see (2.11a)). Since $u_k = 0$ is feasible to problem (2.8), the optimal objective value is at most zero, so (2.11b) is achievable. Clearly, if $W_k \succ 2\theta I$, then (2.11a) holds for any d_k.

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

Model reduction condition. *A search direction* (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 θ *defined in the tangential component condition and a given constant* $\sigma \in (0, 1)$.

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 [5] may benefit by this replacement, but in that method a tangential component is not explicitly computed. Second, the last term on the right-hand side of (2.12) is appropriate because v_k satisfies the Cauchy decrease condition (2.3).

A third condition that will be used in our termination criteria for the primal-dual step computation is the following requirement on the dual residual vector.

Dual residual condition. *The dual residual vector* ρ_k *must satisfy*

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

for a given constant $\kappa \in (0, 1)$.

Inequality (2.13) is similar to the bound commonly enforced in inexact Newton techniques (see [11]), with two differences. First, we disregard the residual r_k as this quantity is implicitly controlled by the tangential component condition. Second, the latter term in the min expression on the right-hand side is included to ensure the limits (3.9) and (3.11) corresponding to convergence toward dual feasibility (see Lemma 3.19).

We are now ready to present our termination tests for an iterative solver applied to the primal-dual system (2.6) that make use of the conditions outlined above. We refer to these tests as sufficient merit function approximation reduction termination tests (SMART tests for short) as in [4, 5].

A step satisfying the first termination test ensures 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* $\pi_k = \pi_{k-1}$*, and the dual residual condition* (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 ensure productive perturbations in the Lagrange multiplier estimates. Inclusion of this test is necessary for the special cases when x_k is a first-order optimal solution or infeasible stationary point for problem (1.1) (see (1.3) and (1.4)) but $g_k + A_k^T\lambda_k \neq 0$, because in these situations we may otherwise require an exact (or nearly exact) solution to the primal-dual system (2.6) to produce an acceptable search direction.

Termination test 2. *If for a given constant* $\epsilon_2 > 0$ *we have*

$$\|A_k^T c_k\| \leq \epsilon_2 \|g_k + A_k^T\lambda_k\|, \tag{2.14}$$

then a step $(d_k, \delta_k) \leftarrow (0, \delta_k)$ *is acceptable if the dual residual condition* (2.13) *holds; i.e.,*

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

for the given $\kappa \in (0,1)$*.*

We restrict consideration of this test to iterations where (2.14) is satisfied so that the algorithm not only reduces dual infeasibility but eventually satisfies one of the other tests so that progress in the primal space will be made once again. 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 $(v_k, u_k, d_k) \leftarrow (0,0,0)$), while only perturbing the multiplier estimates.

The last termination test requires that the normal component v_k yields a reduction in the linear model l_k of the constraints (i.e., it requires $\|c_k\| - \|c_k + A_k v_k\| > 0$) and that the reduction in l_k provided by d_k is proportional to that obtained by v_k. We consider only an increase in the penalty parameter value if this test is satisfied.

Termination test 3. *A search direction* (d_k, δ_k) *is acceptable if the tangential component condition* (2.10) *or* (2.11) *is satisfied, the dual residual condition* (2.13) *holds, and*

$$\|c_k\| - \|c_k + A_k d_k\| \geq \epsilon_3(\|c_k\| - \|c_k + A_k v_k\|) > 0 \tag{2.16}$$

for a given constant $\epsilon_3 \in (0,1)$*.*

For search directions satisfying Termination test 3, we set

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

for a given $\tau \in (0,1)$, which implies (see (1.8))

$$\begin{aligned}\Delta m_k(d_k;\pi_k) \geq\ & \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \tau\pi_k(\|c_k\| - \|c_k + A_k d_k\|) \\ \geq\ & \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \tau\epsilon_3\pi_k(\|c_k\| - \|c_k + A_k v_k\|),\end{aligned}$$

so the model reduction condition (2.12) is satisfied for $\sigma = \tau\epsilon_3$. From now on we assume the constants τ, σ, and ϵ_3 are chosen to satisfy this relationship for consistency between Termination tests 1 and 3.

Now, as in [5], we consider the fact 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 strategy.

HESSIAN MODIFICATION STRATEGY. *Let W_k be the current Hessian approximation, and let a trial search direction (d_k, δ_k) be given. If the tangential component u_k of d_k satisfies* (2.10) *or* (2.11a)*, then maintain the current W_k; otherwise, modify W_k to increase its smallest eigenvalue.*

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 (2.11a).

Finally, once an acceptable search direction has been computed, we perform a backtracking line search on the penalty function $\phi(x; \pi_k)$. We choose α_k such that the sufficient decrease 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 new dual iterate λ_{k+1} we require

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

which can be achieved, for example, by choosing β_k as the solution to

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

and setting $\lambda_{k+1} = \lambda_k + \beta_k \delta_k$, as is done in our implementation described in section 4.

The complete algorithm is the following.

ALGORITHM `TRINS`. Trust Region Inexact Newton with SMART Tests

Choose parameters $0 < \kappa, \epsilon_3, \tau, \eta < 1$, $0 < \omega, \theta, \epsilon_2, \delta_\pi$, and $\psi, \zeta \geq 0$
Initialize x_0, λ_0, and $\pi_{-1} > 0$ and set $v_{-1} \leftarrow 0$, $g_{-1} \leftarrow g_0$, $A_{-1} \leftarrow A_0$, and $\sigma \leftarrow \tau\epsilon_3$
for $k = 0, 1, 2, \ldots$, until termination criteria for (1.1) are satisfied
  Compute f_k, g_k, c_k, A_k, and W_k and initialize $\pi_k \leftarrow \pi_{k-1}$
  Compute v_k satisfying the normal component condition
  Compute an approximate solution (d_k, δ_k) to (2.6) with the unmodified W_k
  repeat
    Set $u_k \leftarrow d_k - v_k$
    if (d_k, δ_k) satisfies Termination test 1 or 3, then **break**
    if (d_k, δ_k) satisfies Termination test 2,
     then set $(v_k, u_k, d_k) \leftarrow (0, 0, 0)$ and **break**
    Run the Hessian modification strategy to update W_k
    Compute an improved solution (d_k, δ_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 $\alpha_k \leftarrow 1$
  Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$ and choose λ_{k+1} yielding (2.19)
endfor

A particular implementation of Algorithm `TRINS` is described in section 4.

3. Algorithm analysis.

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

ASSUMPTION 3.1. *The sequence $\{x_k\}$ generated by Algorithm* TRINS *is contained in a convex set over which the functions f and c and their first derivatives are bounded and Lipschitz continuous. Moreover, the sequence $\{W_k\}$ is bounded over all k, and the iterative linear system solver can make the residual (ρ_k, r_k) arbitrarily small 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 by 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 search direction, i.e., the value for W_k computed after all modifications from the Hessian modification strategy have been performed.

There are a number of iterative linear system 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 section 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 modifications of W_k as in the Hessian modification strategy will eventually produce a consistent system.

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 (1.3)) or at least a stationary point of the infeasibility measure $\varphi(x)$ (see (1.4)). The following lemma formalizes our discussion dealing with the remaining cases.

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) for $j = 1, 2, \ldots$ 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 the residual is eventually made arbitrarily small; i.e., we have

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

We consider a series of cases and show that, in each one, 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) (i.e., (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 by (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. Then, by (2.9) and since $\tilde{A}_k^T$ has the same range space as A_k^T, there exists $\tilde{\delta}_k^j$ for all j with

$$\tilde{A}_k^T\tilde{\delta}_k^j = A_k^T\delta_k^j = -g_k - A_k^T\lambda_k - W_k(v_k + u_k^j) + \rho_k^j. \tag{3.5}$$

Thus, for some γ_a, $\gamma_b > 0$, we have by (3.3) that

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

for all large $j \in J$, so with (3.5) and (3.2b) we find

$$\begin{aligned}
&(g_k + W_kv_k)^Tu_k^j + \tfrac{1}{2}(u_k^j)^TW_ku_k^j \\
&= -\tfrac{1}{2}(u_k^j)^TW_ku_k^j - \lambda_k^TA_ku_k^j - (\tilde{\delta}_k^j)^T\tilde{A}_ku_k^j + (\rho_k^j)^Tu_k^j \\
&= -\tfrac{1}{2}(u_k^j)^TW_ku_k^j - \lambda_k^Tr_k^j - (\tilde{\delta}_k^j)^T\tilde{r}_k^j + (\rho_k^j)^Tu_k^j \\
&\le -\theta\|u_k^j\|^2 + \|\lambda_k\|\|r_k^j\| + (\gamma_a + \gamma_b\|u_k^j\|)\|\tilde{r}_k^j\| + \|\rho_k^j\|\|u_k^j\|.
\end{aligned} \tag{3.6}$$

Let us view the right-hand sides of (3.6) for $j \in J$ as a sequence of concave quadratic functions of $\|u_k^j\|$. By (3.3), each function in this sequence has the form

$$h^j(y) = -ay^2 + b^jy + c^j \tag{3.7}$$

with $a > 0$ and $\lim_{j\to\infty}\{b^j, c^j\} = 0$, 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 (3.6), we have that (2.11b) holds for all large $j \in J$. All together, we have shown that the tangential component condition (2.10) or (2.11) is satisfied for all large j.

Next, we consider the cases $\|A_k^Tc_k\| > 0$ and $\|A_k^Tc_k\| = 0$ and show that in each case the remaining conditions of Termination test 1, 2, or 3 will eventually be satisfied for large j. First, if $\|A_k^Tc_k\| > 0$, then $\|A_kv_k\| > 0$ by (2.3). Since we have assumed that the algorithm has not terminated finitely and so (3.1a) does not hold, this means that the right-hand side of (2.13) is positive. It then follows from (3.3) that the dual residual condition (2.13) is satisfied for all large j. Similarly, by (2.7), (2.9), and (3.3),

$$\begin{aligned}
\frac{\|c_k\| - \|c_k + A_kd_k^j\|}{\|c_k\| - \|c_k + A_kv_k\|} &\ge \frac{\|c_k\| - \|c_k + A_kv_k\| - \|A_ku_k^j\|}{\|c_k\| - \|c_k + A_kv_k\|} \\
&= 1 - \frac{\|r_k^j\|}{\|c_k\| - \|c_k + A_kv_k\|} \ge \epsilon_3
\end{aligned}$$

for all large j, where $\epsilon_3 \in (0, 1)$ is defined for (2.16). Thus, (2.16) 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 we have assumed that (3.1b) does not hold. By (2.9) we then have $\|g_k + A_k^T(\lambda_k + \delta_k^j)\| = \|\rho_k^j - W_k u_k^j\|$ for all j. If $\liminf \|u_k^j\| = 0$, then by (3.3) this quantity is arbitrarily small for some large j. This, our assumption that (3.1a) does not hold, and the fact that the right-hand side of (2.15) is nonzero as $\|g_k + A_k^T\lambda_k\| \neq 0$, together imply that Termination 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 $\pi_k = \pi_{k-1}$ (and using $A_k d_k^j = A_k u_k^j = r_k^j$) we have

$$\begin{aligned}
&-\Delta m_k(d_k^j;\pi_k) + \max\{\tfrac{1}{2}(u_k^j)^T W_k u_k^j, \theta\|u_k^j\|^2\} + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|)\\
&= g_k^T d_k^j - \pi_k(\|c_k\| - \|c_k + A_k d_k^j\|) + \max\{\tfrac{1}{2}(u_k^j)^T W_k u_k^j, \theta\|u_k^j\|^2\}\\
&= g_k^T u_k^j - \pi_k(\|c_k\| - \|c_k + r_k^j\|) + \tfrac{1}{2}(u_k^j)^T W_k u_k^j\\
&\leq -\theta\|u_k^j\|^2 + \|\lambda_k\|\|r_k^j\| + (\gamma_a + \gamma_b\|u_k^j\|)\|\tilde{r}_k^j\| + \|\rho_k^j\|\|u_k^j\| + \pi_k\|r_k^j\|,
\end{aligned} \tag{3.8}$$

where for the last inequality we have applied (3.6). Indeed, the right-hand side of (3.8) differs from that of (3.6) only in the term $\pi_k\|r_k^j\|$. Therefore, as before, the right-hand side of (3.8) can be viewed as a concave quadratic function of the form (3.7), and as argued above it is negative for all large j (note that since $v_k = 0$, the set J now includes 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 search direction (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 α_k satisfying (2.18) if $d_k \neq 0$. By properties of the directional derivative $D\phi_k(d_k;\pi_k)$ of ϕ at x_k along d_k, it is clear that there exists an $\bar{\alpha}_k > 0$ such that (2.18) with $-\Delta m_k(d_k;\pi_k)$ replaced by $D\phi_k(d_k;\pi_k)$ is satisfied for all $\alpha_k \in [0, \bar{\alpha}_k]$ if $D\phi_k(d_k;\pi_k) < 0$. Thus, since $D\phi_k(d_k;\pi_k) \leq -\Delta m_k(d_k;\pi_k) < 0$ (see Lemmas 3.8, 3.9, and 3.6 and (2.7)), 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. *Suppose that Assumption* 3.1 *holds. If all limit points of* $\{A_k\}$ *have full row rank, then* $\{\pi_k\}$ *is bounded and*

$$\lim_{k\to\infty} \left\| \begin{bmatrix} g_k + A_k^T\lambda_{k+1} \\ c_k \end{bmatrix} \right\| = 0. \tag{3.9}$$

Otherwise,

$$\lim_{k\to\infty} \|A_k^T c_k\| = 0, \tag{3.10}$$

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

$$\lim_{k\to\infty} \|g_k + A_k^T\lambda_{k+1}\| = 0. \tag{3.11}$$

We prove Theorem 3.3 with a series of observations and lemmas. For convenience, we define the index set T_1 to denote iterations where Termination test 1 is satisfied,

T_3 to denote iterations where Termination test 3 is satisfied and Termination test 1 is not, and T_2 to denote all iterations where only Termination test 2 is satisfied.

The first result concerns a one-dimensional problem and will be useful for providing a tight bound on the reduction obtained in the model l_k of the constraints by the normal component v_k. Its proof can be found in [6].

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

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

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

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

The proof of the next lemma includes the application of this result to problem (2.2). Note that, in this and Lemma 3.6, we omit consideration of $k \in T_2$ since in such iterations we set $v_k \leftarrow 0$.

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

$$\|c_k\| - \|c_k + A_k v_k\| \ge \gamma_1 \frac{\|A_k^T c_k\|^2}{\|c_k\|}. \tag{3.12}$$

Proof. For $k \notin T_2$, inequality (3.12) clearly holds when $\|A_k^T c_k\| = 0$ since 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

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

Applying Lemma 3.4 to this problem yields

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

Thus, since v_k satisfies the Cauchy decrease condition (2.3), we find with the relation $2a(a-b) \ge a^2 - b^2$ for any $a, b \in \mathbb{R}$ that

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

which proves the result since $\|c_k\| \ne 0$ (as we are considering the case $\|A_k^T c_k\| \ne 0$) and since $\|A_k^T A_k\|$ is bounded under Assumption 3.1. ∎

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*

$$\gamma_2 \|A_k^T c_k\|^2 \leq \|v_k\| \leq \omega \|A_k^T c_k\|, \tag{3.13}$$

and hence v_k is bounded in norm over all k.

Proof. The inequality on the right-hand side of (3.13) 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 (3.13) 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 (3.13) follows trivially. Otherwise, the above inequality and (3.12) yield

$$\|v_k\| \geq \frac{\|c_k\| - \|c_k + A_k v_k\|}{\|A_k\|} \geq \gamma_1 \frac{\|A_k^T c_k\|^2}{\|A_k\| \|c_k\|},$$

and so (3.13) follows from the fact that, under Assumption 3.1, $\|A_k\|$ and $\|c_k\|$ 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 $u_k = 0$. If $k \notin T_2$, then the tangential component condition, (2.10) or (2.11), is satisfied since Termination tests 1 and 3 require it. If (2.10) holds, then u_k is bounded by Lemma 3.6. Otherwise, (2.11) is satisfied, which states

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

and by (2.11a) we then have

$$\theta \|u_k\|^2 - \|u_k\| \|g_k + W_k v_k\| - \zeta \|v_k\| \leq 0. \tag{3.14}$$

The expression on the left-hand side of this inequality is a convex quadratic function in $\|u_k\|$ whose coefficients are bounded for all k (by Assumption 3.1 and Lemma 3.6). Therefore, values $\|u_k\|$ satisfying (3.14) are bounded. ∎

We have thus shown in Lemmas 3.6 and 3.7 that the entire primal search direction $d_k = v_k + u_k$ is bounded in norm over all k.

Next, we turn to a series of results related to the model reduction condition (2.12). The first result illustrates the connection between $\Delta m(d;\pi)$ and the directional derivative of the penalty function ϕ; a proof can be found in [4].

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

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

The results above can be used to obtain a bound for $\Delta m_k(d_k;\pi_k)$. As in Lemmas 3.5 and 3.6, we omit consideration of $k \in T_2$ since in such iterations $v_k \leftarrow 0$.

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. For $k \notin T_2$, the model reduction condition (2.12) and Lemma 3.5 yield

$$\begin{aligned}\Delta m_k(d_k;\pi_k) \geq\ & \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} + \sigma\pi_k(\|c_k\| - \|c_k + A_k v_k\|)\\ \geq\ & \theta\|u_k\|^2 + \sigma\pi_k\gamma_1\frac{\|A_k^T c_k\|^2}{\|c_k\|}.\end{aligned}$$

The result follows as $\|c_k\|$ is bounded under Assumption 3.1. ∎

The squared norm of d_k can be bounded above by a similar quantity.

LEMMA 3.10. *There exists $\gamma_4 \geq 2$ such that, for all k, we have*

$$\|d_k\|^2 \leq \gamma_4\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^T c_k\|^2\right). \tag{3.15}$$

Proof. The result follows trivially for $k \in T_2$ since in such iterations we set $d_k \leftarrow 0$. For $k \notin T_2$, from (2.7) and (3.13) it follows that

$$\begin{aligned}\|d_k\|^2 =\ & \|u_k\|^2 + 2u_k^T v_k + \|v_k\|^2\\ \leq\ & 2\left(\|u_k\|^2 + \|v_k\|^2\right)\\ \leq\ & 2\left(\|u_k\|^2 + \omega^2\|A_k^T c_k\|^2\right),\end{aligned} \tag{3.16}$$

so (3.15) holds for $\gamma_4 = 2\max\{1,\omega^2\}$. ∎

We are now ready to prove the limit (3.10). Here, it will be convenient to work with the scaled and shifted penalty function

$$\tilde\phi(x;\pi) \triangleq \tfrac{1}{\pi}\left(f(x) - f_{\min}\right) + \|c(x)\|, \tag{3.17}$$

where $f_{\min}$ is the infimum of f over the smallest convex set containing the iterates of the algorithm whose existence follows from Assumption 3.1. This function has an interesting monotonicity 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 $\{\tilde\phi(x_k;\pi_k)\}$ is monotonically decreasing.

Proof. First, if $k \in T_2$, then $\tilde\phi(x_{k+1};\pi_{k+1}) = \tilde\phi(x_k;\pi_k)$ and $\Delta m_k(d_k;\pi_k) = 0$ since we set $d_k \leftarrow 0$ and $\pi_{k+1} \leftarrow \pi_k$ for $k \in T_2$, 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

$$\tilde\phi(x_{k+1};\pi_{k+1}) \leq \tilde\phi(x_k;\pi_k) + \left(\tfrac{1}{\pi_{k+1}} - \tfrac{1}{\pi_k}\right)(f_{k+1} - f_{\min}) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k). \tag{3.18}$$

The fact that $\{\pi_k\}$ is a monotonically increasing sequence and the nonnegativity of $(f_{k+1} - f_{\min})$ then yield the result. ∎

The limit (3.10) now follows from the above results.

LEMMA 3.12. *The sequence $\{x_k\}$ yields*

$$\lim_{k\to\infty}\|A_k^T c_k\| = 0.$$

Proof. Consider an arbitrary value $\gamma_5 > 0$, and define the set

$$S = \{x : \gamma_5 \leq \|A(x)^T c(x)\|\}.$$

We prove the result by showing that there can only be a finite number of iterates with $x_k \in S$. Since γ_5 is chosen arbitrarily, this will prove the lemma.

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 $k \geq k'$ since $d_k \leftarrow 0$ for $k \in T_2$, and (2.15) and (2.19) yield

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

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

$$\epsilon_2 \|g_k + A_k^T \lambda_k\| \geq \|A_k^T c_k\| = \|A_{k'}^T c_{k'}\| \geq \gamma_5 > 0.$$

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

$$\|v_k\| \geq \gamma_2 \|A_k^T c_k\|^2 \geq \gamma_2 \gamma_5^2. \tag{3.19}$$

Similarly, by Lemma 3.7 we may define

$$u_{\sup}^S \triangleq \sup\{\|u_k\| : x_k \in S\} < \infty,$$

and it follows from (3.19) that

$$\|u_k\| \leq \left(u_{\sup}^S / (\gamma_2 \gamma_5^2)\right) \|v_k\|. \tag{3.20}$$

By Lemmas 3.6 and 3.9, there exists a constant $\gamma_6 > 0$ such that

$$\begin{aligned} \Delta m_k(d_k; \pi_k) &\geq \gamma_3 \pi_k \|A_k^T c_k\|^2 \\ &\geq \gamma_3 \pi_k \frac{1}{\omega^2} \|v_k\|^2 \\ &\geq \gamma_6 \pi_k \|d_k\|^2, \end{aligned} \tag{3.21}$$

where we used (3.20) to derive the last inequality. Next, we note that if the line search condition (2.18) does not hold for some $\bar{\alpha} > 0$ during iteration k, then

$$\phi(x_k + \bar{\alpha} d_k; \pi_k) - \phi(x_k; \pi_k) > -\eta \bar{\alpha} \Delta m_k(d_k; \pi_k). \tag{3.22}$$

However, with Lemma 3.8, a Taylor expansion of $\phi(x; \pi_k)$ at x_k along d_k yields for some $\gamma_7 > 0$ independent of x_k that

$$\begin{aligned} \phi(x_k + \bar{\alpha} d_k; \pi_k) - \phi(x_k; \pi_k) &\leq \bar{\alpha} D\phi(d_k; \pi_k) + \gamma_7 \bar{\alpha}^2 \pi_k \|d_k\|^2 \\ &\leq -\bar{\alpha} \Delta m_k(d_k; \pi_k) + \gamma_7 \bar{\alpha}^2 \pi_k \|d_k\|^2, \end{aligned} \tag{3.23}$$

and so with (3.22) we have

$$(1 - \eta) \Delta m_k(d_k; \pi_k) < \gamma_7 \bar{\alpha} \pi_k \|d_k\|^2. \tag{3.24}$$

From (3.21) we then find

$$(1 - \eta) \gamma_6 \pi_k \|d_k\|^2 < \gamma_7 \bar{\alpha} \pi_k \|d_k\|^2,$$

which implies $\bar{\alpha} \geq (1-\eta)\gamma_6/\gamma_7$. Since the backtracking line search decreases the steplength α_k by 1/2 after each trial, we have $\alpha_k \geq \alpha_{\min}^S \triangleq (1-\eta)\gamma_6/(2\gamma_7)$. Thus, along with Lemmas 3.9 and 3.11, we have

$$\begin{aligned}\tilde{\phi}(x_{k+1};\pi_{k+1}) \leq\ & \tilde{\phi}(x_k;\pi_k) - \tfrac{1}{\pi_k}\eta\alpha_k\Delta m_k(d_k;\pi_k)\\ \leq\ & \tilde{\phi}(x_k;\pi_k) - \eta\alpha_{\min}^S\gamma_3\|A_k^Tc_k\|^2\\ \leq\ & \tilde{\phi}(x_k;\pi_k) - \eta\alpha_{\min}^S\gamma_3\gamma_5^2,\end{aligned} \tag{3.25}$$

since $x_k \in S$.

For the purpose of deriving a contradiction, suppose there are 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 (3.25) implies a reduction in $\tilde{\phi}$ by at least 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 $x_k \in S$. □

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 the particular case when all limit points of the sequence $\{A_k\}$ produced by Algorithm TRINS have full row rank. That is, we now focus on the situation where the singular values of the constraint Jacobians $\{A_k\}$ remain bounded away from zero for all k sufficiently large. We prove that then the penalty parameter will remain bounded and the sequence $\{(x_k,\lambda_k)\}$ will satisfy the limit (3.9).

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

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

$$\lim_{k\to\infty}\|c_k\| = 0. \tag{3.26}$$

Proof. As the singular values of $\{A_k\}_{k\geq k_A}$ are bounded away from zero, there exists $\gamma_8 > 0$ such that

$$\|A_k^Tc_k\| \geq \gamma_8\|c_k\| \tag{3.27}$$

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 l_k of the constraints will eventually remain large in proportion to the norm of the normal component v_k over all iterations when the search direction satisfies Termination test 3. This result is important as the penalty parameter will be increased only if this test is satisfied.

Lemma 3.15. *Suppose that there exists $k_A \geq 0$ such that the singular values of $\{A_k\}_{k\geq k_A}$ are bounded away from zero. Then, for all $k \geq k_A$ such that $k \in T_3$, there exists $\gamma_9 > 0$ such that*

$$\|v_k\| \leq \gamma_9(\|c_k\| - \|c_k + A_kd_k\|). \tag{3.28}$$

Proof. As the singular values of $\{A_k\}_{k\geq k_A}$ are bounded away from zero, (3.27) holds for all $k \geq k_A$. Along with Lemma 3.5, this implies that for $\gamma_{10} = \gamma_1\gamma_8^2 > 0$ we have

$$\|c_k\| - \|c_k + A_kv_k\| \geq \gamma_1\frac{\|A_k^Tc_k\|^2}{\|c_k\|} \geq \gamma_{10}\|c_k\|,$$

so along with (3.13) and (2.16) we find

$$\begin{aligned}\|v_k\| \leq\ & \omega\|A_k^T\|\|c_k\| \\ \leq\ & \omega\|A_k^T\|\tfrac{1}{\gamma_{10}}(\|c_k\| - \|c_k + A_k v_k\|) \\ \leq\ & \omega\|A_k^T\|\tfrac{1}{\gamma_{10}\epsilon_3}(\|c_k\| - \|c_k + A_k d_k\|).\end{aligned}$$

Thus, (3.28) holds as $\|A_k^T\|$ is bounded under Assumption 3.1. ∎

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

LEMMA 3.16. *Suppose that there exists $k_A \geq 0$ such that the singular values of $\{A_k\}_{k \geq k_A}$ are bounded 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 to consider here iterations where $k \in T_3$ since π_k remains unchanged otherwise. That is, we may assume the inequality (2.16) holds and that the tangential component condition (2.10) or (2.11) is satisfied. We show below that there exists $\gamma_{11} > 0$ such that

$$g_k^T d_k + \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} \leq \gamma_{11}\left(\|c_k\| - \|c_k + A_k d_k\|\right), \tag{3.29}$$

and so from (2.17) we have that $\{\pi_k^{trial}\}_{k \in T_3}$ is bounded. This, along with the fact that when Algorithm TRINS increases π, it does so by at least δ_π, proves the result.

To prove (3.29), we first consider the case where $\|u_k\| \leq \psi\|v_k\|$, so with (3.16) we have

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

Then, by Lemmas 3.6 and 3.15, and since $\|g_k\|$ and $\|W_k\|$ are bounded under Assumption 3.1, there exist $\gamma_{12}, \gamma_{12}', \gamma_{12}'' > 0$ such that

$$\begin{aligned}g_k^T d_k + \max\{\tfrac{1}{2}u_k^T W_k u_k, \theta\|u_k\|^2\} \leq\ & g_k^T d_k + \gamma_{12}\|u_k\|^2 \\ \leq\ & \|g_k\|\|d_k\| + \gamma_{12}\psi^2\|v_k\|^2 \\ \leq\ & \|g_k\|\sqrt{2(\psi^2+1)}\|v_k\| + \gamma_{12}\psi^2\|v_k\|^2 \\ \leq\ & \gamma_{12}'\|v_k\| \\ \leq\ & \gamma_{12}''(\|c_k\| - \|c_k + A_k d_k\|).\end{aligned}$$

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 $\gamma_{13}, \gamma_{13}' > 0$ such that

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

Thus, (3.29) follows with $\gamma_{11} = \max\{\gamma_{12}'', \gamma_{13}'\}$. ∎

We have thus completed our presentation of results specifically for the case 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 regardless of the ranks of these limit points.

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 away from zero.*

Proof. First, if $k \in T_2$, then $d_k \leftarrow 0$ and Algorithm `TRINS` sets $\alpha_k \leftarrow 1$. It remains to consider $k \notin T_2$.

As in the proof of Lemma 3.12 (see (3.22)–(3.24)), we have for $k \notin T_2$ that if (2.18) fails for $\bar{\alpha} > 0$, then

$$(1-\eta)\Delta m_k(d_k;\pi_k) < \bar{\alpha}\gamma_7\pi_k\|d_k\|^2.$$

Lemmas 3.9 and 3.10 then yield

$$(1-\eta)\gamma_3\left(\|u_k\|^2 + \pi_k\|A_k^Tc_k\|^2\right) < \bar{\alpha}\gamma_4\gamma_7\pi_k\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^Tc_k\|^2\right),$$

so

$$\bar{\alpha} > \frac{(1-\eta)\gamma_3\left(\|u_k\|^2 + \pi_k\|A_k^Tc_k\|^2\right)}{\gamma_4\gamma_7\pi_k\left(\|u_k\|^2 + \max\{1,\pi_k\}\|A_k^Tc_k\|^2\right)} \geq \alpha_{\min},$$

where the constant $\alpha_{\min} > 0$ 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 the proof of Lemma 3.17, if $k \notin T_2$, there exists $\alpha_k \geq \alpha_{\min}/2$ satisfying (2.18). Thus, since $d_k \leftarrow 0$ and $\Delta m_k(d_k;\pi_k) = 0$ for $k \in T_2$, there exists $\gamma_{14} > 0$ such that

$$\phi(x_k;\pi_k) - \phi(x_k + \alpha_k d_k;\pi_k) \geq \gamma_{14}\Delta m_k(d_k;\pi_k)$$

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

$$\begin{aligned}\phi(x_{\bar{k}};\bar{\pi}) - \phi(x_k;\bar{\pi}) &= \sum_{j=\bar{k}}^{k-1}\left(\phi(x_j;\bar{\pi}) - \phi(x_{j+1};\bar{\pi})\right)\\ &\geq \gamma_{14}\sum_{j=\bar{k}}^{k-1}\Delta m_j(d_j;\bar{\pi})\\ &\geq \gamma_{14}\gamma_3\sum_{j=\bar{k},j\notin T_2}^{k-1}\left(\|u_j\|^2 + \bar{\pi}\|A_j^Tc_j\|^2\right).\end{aligned}$$

If there are only a finite number of iterates with $k \notin T_2$, then the result follows as $d_k \leftarrow 0$ for $k \in T_2$. Otherwise, since $\phi(x;\bar{\pi})$ is bounded below under Assumption 3.1, we deduce from the above inequality that

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

The result follows from this limit and Corollary 3.13. ∎

The above results 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. Termination tests 1, 2, and 3 all require that the dual residual condition (2.13) holds for the given $\kappa \in (0,1)$. (Recall that $v_k \leftarrow 0$ and $d_k \leftarrow 0$ for $k \in T_2$, so (2.15) reduces to (2.13).) Thus, under Assumption 3.1, we find from (2.19) and the first block equation of (2.9) that there exist $\gamma_{15}, \gamma'_{15} > 0$ such that

$$\begin{aligned}
\|g_k + A_k^T \lambda_{k+1}\| &\leq \|g_k + A_k^T(\lambda_k + \delta_k)\| \\
&= \|\rho_k - W_k d_k\| \\
&\leq \|\rho_k\| + \gamma_{15}\|d_k\| \\
&\leq \kappa \left\| \begin{bmatrix} g_{k-1} + A_{k-1}^T \lambda_k \\ A_{k-1} v_{k-1} \end{bmatrix} \right\| + \gamma_{15}\|d_k\| \\
&\leq \kappa \|g_{k-1} + A_{k-1}^T \lambda_k\| + \gamma'_{15} \max\{\|v_{k-1}\|, \|d_k\|\}.
\end{aligned}$$

Consider an arbitrary $\gamma_{16} > 0$. Corollary 3.13 and Lemma 3.18 imply that there exists $k' > 0$ such that for all $k \geq k'$ we have $\gamma'_{15} \max\{\|v_{k-1}\|, \|d_k\|\} \leq (1-\kappa)\gamma_{16}/2$. Suppose $k \geq k'$ and $\|g_{k-1} + A_{k-1}^T \lambda_k\| > \gamma_{16}$. Then, since $\kappa \in (0,1)$, the above inequality yields

$$\begin{aligned}
(3.30) \quad \|g_k + A_k^T \lambda_{k+1}\| &\leq \kappa\|g_{k-1} + A_{k-1}^T \lambda_k\| + \tfrac{1-\kappa}{2}\gamma_{16} \\
&= (\kappa - 1)\|g_{k-1} + A_{k-1}^T \lambda_k\| + \|g_{k-1} + A_{k-1}^T \lambda_k\| + \tfrac{1-\kappa}{2}\gamma_{16} \\
&< (\kappa - 1)\gamma_{16} + \|g_{k-1} + A_{k-1}^T \lambda_k\| + \tfrac{1-\kappa}{2}\gamma_{16} \\
&= \|g_{k-1} + A_{k-1}^T \lambda_k\| - \tfrac{1-\kappa}{2}\gamma_{16}.
\end{aligned}$$

Therefore, $\{\|g_k + A_k^T \lambda_{k+1}\|\}$ decreases monotonically by at least a constant amount for $k \geq k'$ while $\{\|g_k + A_k^T \lambda_{k+1}\|\} > \gamma_{16}$, so we eventually find $\|g_k + A_k^T \lambda_{k+1}\| \leq \gamma_{16}$ for some $k = k'' > k'$. Then, for $k > k''$ we find from (3.30) that

$$\|g_k + A_k^T \lambda_{k+1}\| \leq \kappa\gamma_{16} + \tfrac{1-\kappa}{2}\gamma_{16} = \tfrac{\kappa+1}{2}\gamma_{16} \leq \gamma_{16}.$$

Since $\gamma_{16} > 0$ was chosen arbitrarily, the result follows. ∎

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

Proof of Theorem 3.3. If all limit points of $\{A_k\}$ have full row rank, then there exists $k_A \geq 0$ such that the singular values of $\{A_k\}_{k \geq k_A}$ are bounded 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 (3.9).

If a limit point of $\{A_k\}$ does not have full row rank, then we have the limit (3.10) by Lemma 3.12. Moreover, if $\{\pi_k\}$ is bounded, then the fact that if Algorithm TRINS increases π_k, it does so by at least δ_π implies that the penalty parameter is in fact constant for all k sufficiently large. This along with Lemma 3.19 yields the limit (3.11). ∎

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 and present three sets of results on a standard test set.

4.1. Implementation details. We implemented the following termination criteria for the `for` loop of the algorithm:

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

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

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

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

where

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

If condition (4.1a) is satisfied, then an optimal solution has been found within the prescribed tolerances. If (4.1b) or (4.1c) is satisfied but (4.1a) is not, then the problem is deemed locally infeasible. We use a relatively large value for the tolerance parameter $\epsilon_{\text{inf},2}$ (compared to the value for $\epsilon_{\text{inf},1}$) as the algorithm may become numerically unstable if π becomes very large, which would make the latter condition in (4.1c) difficult to satisfy for a tighter tolerance. Finally, if (4.1d) is satisfied before any of (4.1a)–(4.1c), then we terminate due to a limit on the number of outer iterations.

Notice that, by using the conditions in (4.1), the algorithm may return a "false infeasible," i.e., a message that the problem is locally infeasible when further iterations may produce an iterate satisfying (4.1a). In practice one may, for example, utilize a smaller value for $\epsilon_{\text{inf},1}$ and/or a larger value for $\pi_{\max}$ to ensure that such a result is not returned, but in our tests the ratio $\epsilon_{\text{prim}}/\epsilon_{\text{inf},1} = 10$ is justified by the fact that prior to the employment of our approach we prescale the objective and constraint functions so that at the starting point the first derivative of each has a ℓ_∞-norm less than or equal to 10; see [28]. Indeed, no "false" declaration of infeasibility is reported in the tests described below.

In terms of the input parameters required in our approach, we direct the reader to [4, 5] for general guidelines for the selection of these values. In particular, the inexactness parameters κ, ϵ_2, and ϵ_3 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 search direction 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 choose $\zeta \leftarrow 0.1$.

Our theoretical results consider a constant ω, but in practice this value should be chosen to reflect the conditioning of the constraint Jacobian during each iteration. Thus, one may set it as a large 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.

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

TABLE 4.1
Parameter values used for Algorithm TRINS.

Parameter	Value	Parameter	Value
ψ	10^{-1}	θ	10^{-12}
κ	10^{-4}	ϵ_2	10^{-2}
ϵ_3	$1 - 10^{-4}$	δ_π	10^{-4}
τ	10^{-1}	ζ	10^{-1}
η	10^{-8}	π_{-1}	10^{-1}
ω_0	10^3	σ	$\tau\epsilon_3$

As mentioned in section 2, an efficient method for the normal step computation, i.e., a method for the approximate solution of problem (2.2), is the CG algorithm with the Steihaug stopping tests [25]. A conventional implementation of this approach, however, can become numerically unstable if A_k is ill-conditioned or rank-deficient, so the computation of v_k in our implementation is performed with an adapted version of the LSQR algorithm of Paige and Saunders [20]. This approach is mathematically equivalent to CG but has better numerical properties when applied to (nearly) singular systems. We extended the implementation in [21] so that the iteration is terminated if a direction of zero curvature is found (at which point the current LSQR iterate is returned), the trust region constraint is violated (at which point the point at which the boundary is crossed is returned as the solution), or a solution v_k satisfying $\|c_k + A_k v_k\| \leq 10^{-4}\|c_k\|$ is computed.

The primal-dual step computation of Algorithm TRINS is performed with an adapted version of the implementation in [22] of the Minimum Residual (MINRES) method [19]. For most runs of Algorithm TRINS performed in our experiments, this code is sufficient for computing acceptable search directions (i.e., directions satisfying Termination test 1, 2, or 3) despite the fact that a preconditioner is not implemented in our code. Occasionally, however, unpreconditioned MINRES is not able to provide a step satisfying one of our termination criteria before the limit of $2(n+t)$ iterations is reached. In such cases we simply accept the last iterate computed by the solver.

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 setting the initial guess for the solution of (2.6) as the zero vector. If this initial vector does not satisfy a termination test, then MINRES iterations are performed until either an acceptable search direction 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 allow only 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 [28].

The primal-dual step computation is summarized in the following algorithm, which covers the repeat loop of Algorithm TRINS. Note that for Termination tests 1 and 3 we use the norm of the entire residual (ρ_k, r_k) on the left-hand side of (2.13) with the expectation that this choice will yield fast local convergence.

ALGORITHM TRINS-STEP. STEP COMPUTATION FOR ALGORITHM TRINS

```
Set j ← 0, (d^0, δ^0) ← (0, 0), W_k ← ∇²_xx L_k, and choose μ > 0
if termination test 1, 2, or 3 is satisfied, return (d_k, δ_k) ← (d^j, δ^j)
while j < 2(n + t)
  Increment j ← j + 1
  Perform a MINRES iteration on (2.6) to compute (d^j, δ^j)
  if j ≥ (n + t)/2 or (2.13) holds for κ = 1/2
    if Termination test 1 or 3 is satisfied, then break
    if Termination test 2 is satisfied, then set (v_k, u_k, d^j) ← (0, 0, 0) and break
    if (2.10) and (2.11a) do not hold for (d^j, δ^j)
      Set j ← 0, (d^0, δ^0) ← (d^j, δ^j), W_k ← W_k + μI, and choose μ > 0
    endif
  endif
endwhile
return (d_k, δ_k) ← (d^j, δ^j)
```

4.2. Numerical experiments. Algorithm TRINS is designed for very large applications, but its effectiveness for handling (nearly) rank-deficient Jacobian matrices can be illustrated on problems of any size. We admit that by testing small problems we do not illustrate the benefits of a matrix-free approach, but such experiments require a more sophisticated implementation in production software, which is out of the scope of this paper.

We tested our MATLAB implementation on a variety of problems in the CUTEr [2, 15] collection. From this set, we selected all of the equality constrained problems for which AMPL [14] models were available [1] 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 two 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 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 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 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 [5] on all three sets of problems. The algorithm described in [5], which we refer to as Algorithm INS, attempts to compute a search direction by (approximately) solving the Newton system (1.6) and so has no features for dealing with Jacobian singularity. The goal is to illustrate that Algorithm TRINS performs well on well-conditioned problems and does not falter when faced with the obstacles posed by rank-deficient constraint formulations that can force other methods to fail.

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 termination messages and outer iteration counts for Algorithms TRINS and INS.

Table 4.2

Termination results for Algorithms TRINS *and* INS *(described in* [5]*) for problems from the CUTEr collection. The original problems are solved along with related models where the constraint set has been perturbed.*

			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	11	opt	6	opt	6
aug3dc	3873	1000	opt	4	opt	5	π	10	opt	4	opt	6
aug3d	3873	1000	opt	3	opt	4	π	10	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	11	opt	11	opt	11	—	—
bt3	5	3	opt	3	opt	6	π	6	opt	2	—	—
bt4	3	2	opt	14	opt	44	—	—	opt	10	—	—
bt5	3	2	opt	7	opt	7	π	7	opt	7	—	—
bt6	5	2	opt	9	opt	17	π	17	opt	9	—	—
bt7	5	3	opt	26	opt	24	π	28	opt	21	—	—
bt8	5	2	opt	10	opt	7	inf	89	opt	10	—	—
bt9	4	2	opt	24	opt	17	π	12	opt	67	—	—
byrdsphr	3	2	opt	21	—	—	—	—	opt	30	—	—
catena	32	11	opt	21	—	—	π	180	opt	31	—	—
dixchlng	10	5	opt	10	opt	11	π	19	opt	10	—	—
dtoc1l	14985	9990	opt	8	opt	8	π	8	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	11	opt	9	opt	9
dtoc1nd	735	490	opt	60	opt	57	π	23	—	—	—	—
dtoc2	5994	3996	opt	78	opt	18	π	17	opt	16	opt	31
dtoc3	14996	9997	opt	20	opt	17	inf	33	opt	3	—	—
dtoc4	14996	9997	opt	32	opt	32	inf	20	opt	3	opt	3
dtoc5	9998	4999	opt	22	opt	11	inf	24	opt	4	opt	8
dtoc6	10000	5000	opt	63	opt	16	π	13	opt	20	opt	27
eigena2	110	55	opt	23	opt	21	inf	44	opt	18	opt	19
eigenaco	110	55	opt	4	opt	5	inf	16	opt	4	opt	4
eigenb2	110	55	opt	14	opt	26	—	—	opt	28	opt	15
eigenbco	110	55	opt	103	—	—	—	—	—	—	opt	65
eigenc2	462	231	opt	21	opt	22	π	22	opt	20	—	—
eigencco	30	15	opt	16	opt	13	π	35	opt	16	—	—
fccu	19	8	opt	3	opt	2	inf	11	opt	3	—	—
genhs28	10	8	opt	3	opt	3	inf	12	opt	3	opt	3
gilbert	1000	1	opt	44	opt	31	opt	44	opt	20	—	—

Table 4.3

Termination results for Algorithms TRINS *and* INS *(described in* [5]*) for problems from the CUTEr collection. The original problems are solved along with related models where the constraint set has been perturbed.*

			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	9	opt	8	π	10	opt	4	opt	9
hager1	10000	5000	opt	4	opt	5	—	—	opt	3	opt	12
hager2	10000	5000	opt	2	opt	4	opt	4	opt	2	—	—
hager3	10000	5000	opt	2	opt	4	opt	4	opt	2	—	—
hs006	2	1	opt	2	opt	15	inf	17	opt	2	—	—
hs007	2	1	opt	24	opt	14	π	13	opt	287	—	—
hs009	2	1	opt	3	opt	3	inf	11	opt	4	opt	4
hs026	3	1	opt	17	opt	35	π	9	opt	14	—	—
hs027	3	1	opt	18	opt	48	π	26	opt	79	—	—
hs028	3	1	opt	1	opt	1	inf	10	opt	1	opt	1
hs039	4	2	opt	24	opt	17	π	12	opt	67	—	—
hs040	4	3	opt	4	opt	4	π	8	opt	4	—	—
hs046	5	2	opt	18	opt	21	π	21	opt	14	—	—
hs047	5	3	opt	16	opt	43	—	—	—	—	—	—
hs048	5	2	opt	1	opt	1	inf	10	opt	1	opt	1
hs049	5	2	opt	12	opt	12	π	12	opt	12	opt	12
hs050	5	3	opt	8	opt	8	π	13	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	11	π	10	opt	29	—	—
hs077	5	2	opt	9	opt	34	π	77	opt	9	—	—
hs078	5	3	opt	5	opt	6	π	52	opt	4	opt	244
hs079	5	3	opt	4	opt	7	π	5	opt	4	—	—
hs100lnp	7	2	opt	7	opt	38	π	30	opt	8	—	—
hs111lnp	10	3	opt	63	—	—	π	27	opt	766	—	—
lch	600	1	opt	62	opt	8	inf	8	opt	29	opt	6
maratos	2	1	opt	6	opt	7	π	8	opt	5	—	—
mwright	5	3	opt	7	opt	7	π	11	opt	8	—	—
orthrdm2	4003	2000	opt	12	opt	6	π	6	opt	7	opt	7
orthrds2	203	100	opt	67	opt	34	π	31	opt	78	—	—
orthrega	517	256	opt	95	opt	37	π	42	opt	51	opt	50
orthregb	27	6	opt	2	opt	3	inf	4	opt	2	—	—
orthregc	10005	5000	opt	28	opt	11	π	12	opt	15	opt	20
orthregd	10003	5000	opt	20	opt	7	π	8	opt	9	opt	8
orthrgdm	10003	5000	opt	27	opt	8	π	7	opt	10	opt	8
orthrgds	10003	5000	opt	59	—	—	—	—	opt	32	opt	30
robot	14	9	opt	40	—	—	π	14	opt	6	opt	10

The 'original' columns correspond to results for the original models, the 'perturbed' columns correspond 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 `---`. We also find that an 'opt' message was returned for a couple of infeasible problems. This occurs in our implementation since we prescale the problem using derivative values at the initial point, and in those cases this procedure scales down the infeasible constraint so that a point satisfying (4.1a) exists.

The results suggest that Algorithm `TRINS` is versatile and robust and can successfully be applied to nonconvex problems with (nearly) 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, whereas this is unlikely to be the case for Algorithm `INS`.

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 SMART tests developed in [4, 5] to problems where the constraint Jacobians may lose rank during the solution process. A crucial aspect of the approach is the 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 an infeasibility measure. We close by discussing practical issues about the implementation of our approach.

A drawback of the algorithm described in this paper is that it requires the inexact solution of the normal subproblem (2.2) in addition to the primal-dual system (2.6). A more efficient implementation may be to first attempt to compute a step with the Newton system (1.6) using the termination criteria described in [5] 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). For (nearly) singular systems, an alternative iterative linear system solver is `MINRESQLP`, recently developed by Choi [9], which is a more numerically stable version of the `MINRES` algorithm.

Finally, the algorithm in this paper and those in [4, 5] require further experimentation on large-scale applications, and the extensions that are necessary to apply them to problems with inequality constraints need to be investigated. These issues are addressed in [10].

Acknowledgments. The authors would like to thank the referees whose comments have helped us greatly to improve the presentation.

REFERENCES

[1] H. Y. Benson, *Cute Models*, http://www.orfe.princeton.edu/~rvdb/ampl/nlmodels/cute/index.html (2008).

[2] I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint, *CUTE: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[3] J. V. Burke, *A robust trust region method for constrained nonlinear programming problems*, SIAM J. Optim., 2 (1992), pp. 324–347.

[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] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact Newton method for nonconvex equality constrained optimization*, Math. Program., Ser. A, 2009.

[6] R. H. Byrd, J.-Ch. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.

[7] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Steering exact penalty methods for nonlinear programming*, Optim. Methods Soft., 23 (2008), pp. 197–213.

[8] M. R. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., 1984, SIAM, Philadelphia, PA, (1985), pp. 71–82.

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

[10] F. E. Curtis, O. Schenk, and A. Wächter, *An Interior-point Algorithm for Large-Scale Nonlinear Optimization with Inexact Step Computations*, Technical report RC 24736, IBM T. J. Watson Research Center, Yorktown Heights, NY, 2009.

[11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[12] J.-P. Dussault, *Numerical stability and efficiency of penalty algorithms*, SIAM J. Numer. Anal., 32 (1995), pp. 296–317.

[13] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., J. Wiley and Sons, Chichester, England, 1987.

[14] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, 2nd ed., Brooks/Cole, Pacific Grove, CA, 2002.

[15] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[16] M. Heinkenschloss and D. Ridzal, *An inexact trust-region SQP methods with applications to PDE-constrained optimization*, in Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2007, the 7th European Conference on Numerical Mathematics and Advanced Applications, Graz, Austria, Heidelberg, Germany, K. Kunisch, O. Steinbach, and G. Of, eds., Springer-Verlag, Berlin, 2007, pp. 613–620.

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

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

[19] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[20] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[21] C. C. Paige and M. A. Saunders, *LSQR: Sparse Equations and Least Squares*, http://www.stanford.edu/group/SOL/software/lsqr.html (2002).

[22] C. C. Paige and M. A. Saunders, *MINRES: Sparse Symmetric Equations*, http://www.stanford.edu/group/SOL/software/minres.html (2004).

[23] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Program., 49 (1990), pp. 189–213.

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

[25] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[26] P. L. Toint, *Nonlinear Stepsize Control, Trust Regions and Regularizations for Unconstrained Optimization*, Technical report TR08-15, Department of Mathematics, FUNDP, University of Namur, Namur, Belgium, 2008.

[27] S. Ulbrich, *Generalized SQP-methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization*, in Real-Time PDE-Constrained Optimization, L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, eds., SIAM, Philadelphia, PA, 2007, pp. 145–168.

[28] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., Ser. A, 106 (2006), pp. 25–57.

[29] H. Yamashita and T. Tanabe, *A Primal-dual Exterior point Method for Nonlinear Optimization*, http://www.msi.co.jp/nuopt/materials/files/EP.pdf (2008).